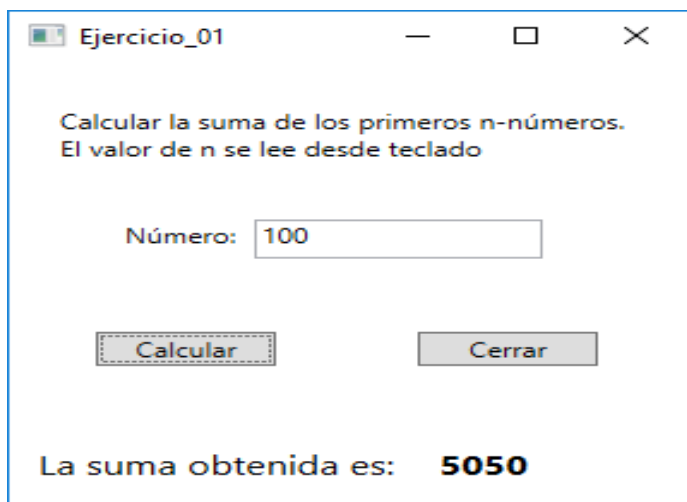


RELACIÓN DE PROBLEMAS Y EJERCICIOS.

Relación Nº: 9

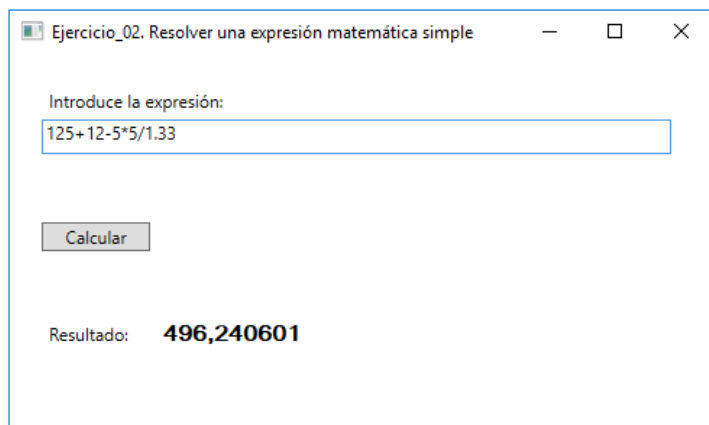
1. Haz un programa que calcule y visualice la suma de los N primeros números enteros. Siendo N un número que se obtiene del teclado.



En esta ocasión el resultado se mostrará al pie de la ventana destacándolo del resto del texto.

Como siempre simplemente he escrito 'a dedo' el código del evento Click de los botones, el resto lo ha construido el Visual C# mientras que yo he ido arrastrando objetos a la ficha.

2. Haz un programa que use una clase que permita evaluar una expresión matemática simple con las operaciones; +, -, \*, / (sumar, restar, multiplicar y dividir) y muestre el resultado.



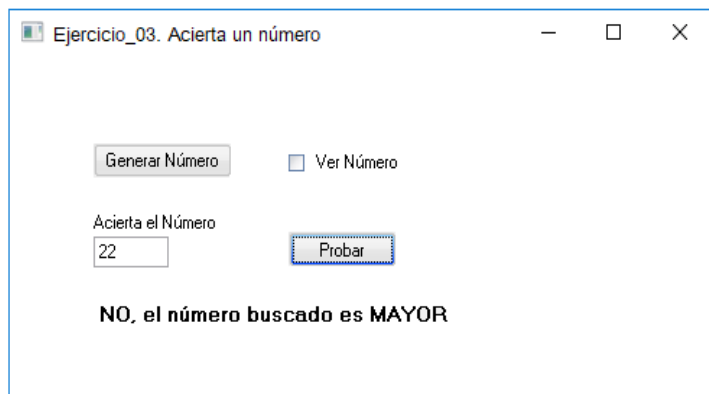
Si la expresión introducida no es correcta se lanza una excepción **ExpresionMalFormadaException** que debemos de capturar en la clase principal.

La expresión se pone completa y al pulsar el botón Calcular se muestra el resultado de evaluar la expresión de izquierda a derecha, no tiene en cuenta las prioridades de los operadores.

Un interfaz podría ser así:

El TextBox solo acepta: números, la coma para los decimales, las cuatros operaciones básicas y la tecla de borrado.

3. Queremos hacer un programa juego, 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 numero 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.

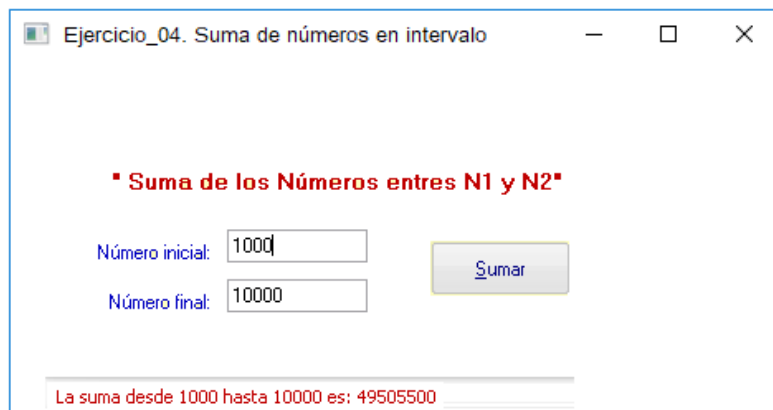
Comienza preparando los controles que necesitas en una ficha nueva como puedes ver en la imagen. En este caso hay un nuevo control que es “Ver Número”, es un CheckBox que se usa para mostrar o no el número generado aleatoriamente y comprobar que todo funciona bien. Es importante que el ejemplo esté bien desarrollado e implementado para ello debes de tener presente

algunas cuestiones:

- Al iniciar el ejemplo, no se puede picar en el botón “Probar” 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.

*Como verás este ejemplo es también fácil pero tenemos que cuidar algunos detalles. A partir de este ejemplo procura usar tu imaginación para hacer el resto de los ejemplos con lógica y rigor.*

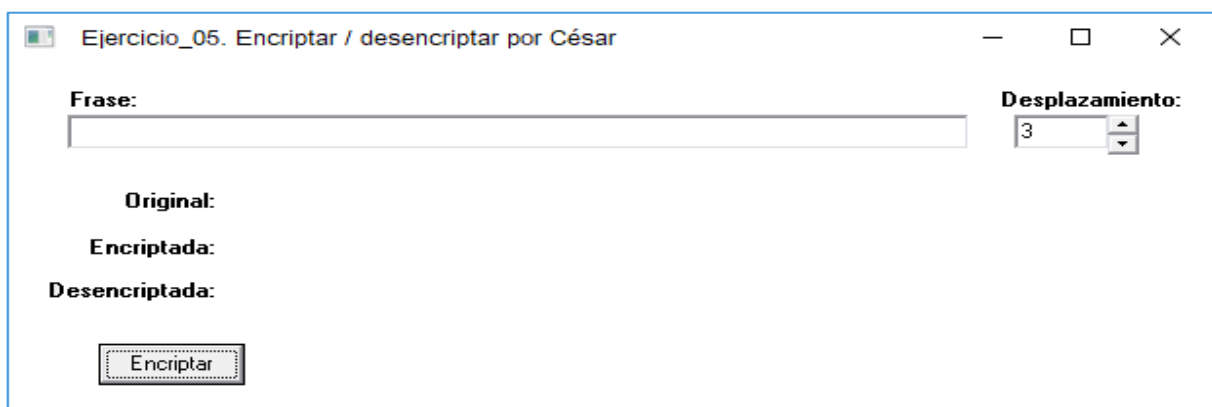
4. Dados dos números por el usuario, (el primero mayor que el segundo) calcular la suma de todos los números comprendidos entre ambos, estos incluidos, y muestra el resultado de la suma en pantalla. Planteamiento: Suponiendo  $N1=5$  y  $N2=9$ ;  $SUMA = 5+6+7+8+9$



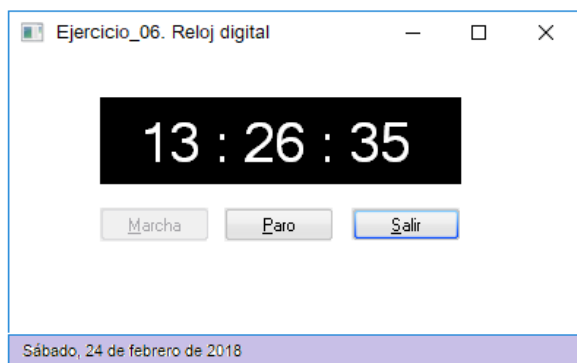
Comienzo preparando una vez más los controles que necesito en una ficha nueva como puedes ver en la imagen. En esta ocasión el resultado lo vamos a mostrar en una barra de estado, concretamente en un “toolStripStatusLabel1”, modificando su propiedad “text”, contenido en el control “StatusStrip” que está en el Cuadro de Herramientas en el apartado “Menús y Barra de Herramientas”. 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, aunque aún con carácter general, y aplicar los criterios comentados al final del ejercicio anterior.

5. 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. Amplíalo para encriptar/desencriptar en ficheros de texto.



6. 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) que su cometido es obvio 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”, como verás en la imagen, esto se consigue accediendo a `DateTime.Now` y a sus métodos, propiedad, etc., usa la ayuda si tienes algún problema para saber cuál de ellos usarás para mostrar el día, la hora, etc.



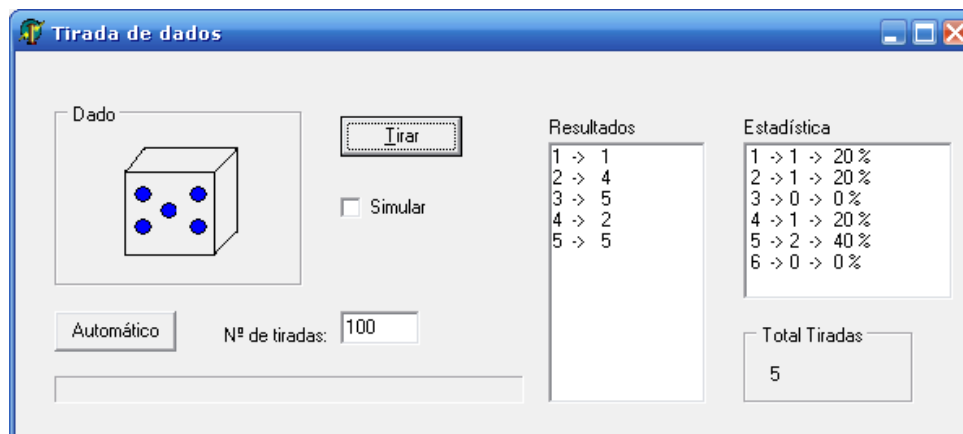
Ampliación: Ponle también lo necesario para que permita poner varias alarmas a una fecha y hora determinada y que suene por los altavoces un zumbido. La lista de las alarmas se puede modificar.

7. Realiza un programa en Windows que lea por teclado el contenido de dos arrays de números enteros y de dimensiones 3\*1 y 1\*3, respectivamente. A continuación se almacenará en un tercer array la matriz resultado de multiplicar las dos primeras. Recuerda que dos matrices, Matriz1 y Matriz2, se multiplican del siguiente modo:

$$\begin{array}{|c|} \hline \text{Matriz1} \\ \hline i \\ \hline k \\ \hline m \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline \text{Matriz2} \\ \hline U & v & w \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline \text{Matriz Resultado} \\ \hline i*u & i*v & i*w \\ \hline k*u & k*v & k*w \\ \hline m*u & m*v & m*w \\ \hline \end{array}$$

Finalmente se visualizará en pantalla el contenido del tercer array. El interfaz gráfico para este ejercicio, y los posteriores si no se indica otra cosa, lo dejo a tu elección, procura que sea “amigable”, sin demasiado colores fuertes.

8. Haz un programa en C# que tenga dos paneles o partes y haz en cada panel un apartado de los siguientes:
- Se le introduzca una frase y muestre la frase PALÍNDROMO o NO PALÍNDROMO según cada caso. También debe de disponer de un Combo en el que haya varias frases que sean palíndromo y pulsando sobre una de ellas se añada al cuadro de texto en vez de escribirla a mano.
  - Calcule y visualice la lista de los números primos menores o iguales que el número entero positivo que se introducirán desde el teclado.
9. Haz una aplicación que permita comprobar que la probabilidad de obtener cualquier valor de la tirada de un dado es la misma.

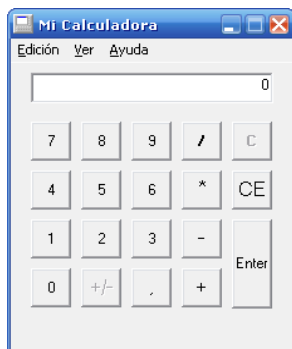


El interfaz es el mostrado en la figura adjunta y permite hacer:

- 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.

10. Haz una aplicación “Mi Calculadora”, que implemente una sencilla calculadora, que permita hacer cálculos básicos de operaciones aritméticas. El intefaz es el mostrado y debe de realizar, o poseer lo siguiente:



- Un Menú principal que permita copiar, cortar y pegar en el cuadro del resultado.
- Un menú secundario que tenga las mismas opciones que el menú principal.
- Los datos y el resultado se muestra a la derecha del cuadro de resultados
- La ventana no se puede maximizar ni redimensionar
- Etc. Usa la imaginación.

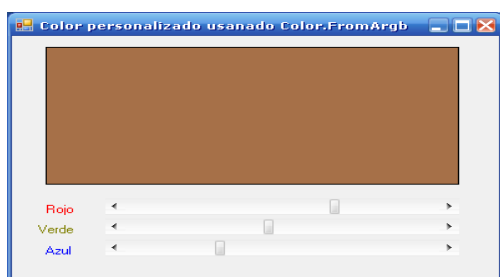
Operaciones de las teclas:

/	Dividir
*	Multiplicar
-	Resta
+	Sumar
C	Borra la última operación

CE	Borra Todo
,	Separador decimal
+/-	Signo unitario para anteponer a un número
Enter	Muestra el resultado de la operación

11. Haz un programa que cuente con varias etiquetas con una frase cada una y que al pulsar el ratón sobre ellas cambie de aspecto, color, tipo de fuente, etc. Este programa debe tener una opción “configuración” donde poder elegir los parámetros de configuración de las frases tanto inicial como el aspecto que se mostrará al pasar el ratón.

12. 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 HScrollBar (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 FromArgb(int r, int v, int a) de la clase color.

13. Haz un pequeño editor de texto de formato RTF parecido al bloc de notas, pero, que permita manipulación de formatos de letras, colores, tipos etc. Usa un menú principal, un menú contextual y una barra de herramientas para las distintas opciones, así como, una barra de estado para mostrar información.

14. Haz un programa que pida por pantalla el nombre y ruta de un fichero de texto y cuente el número de líneas que hay, su tamaño, sus atributos y el número de palabras.

15. Haz un programa que use un fichero binario para manejar datos personales (DNI, edad, sueldo, Apellidos y Nombre) de los clientes. Las opciones que debe de tener son: crear fichero, abrir fichero, editar una ficha, modificar una ficha, ver por pantalla un listado de clientes, etc.

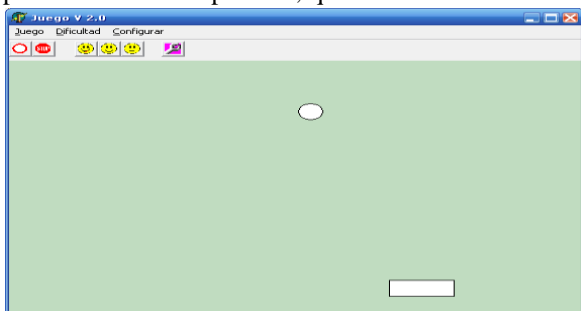
16. Realizar una aplicación Windows que implemente el juego de los dados llamado ‘CRAPS’ que fue en el que Leonardo DiCaprio ganó los pasajes para el Titanic, según cuenta la película. Las reglas del juego son:

*Un jugador tira dos dados. Cada dado tiene seis caras con los valores puntuables 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.

Usa la POO para realizar este ejercicio y haz un método “DemoCraps” que simule un juego completo y muestre los resultados de cada tirada, así como, el número de la tirada.

17. 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 pare 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.



18. El objetivo de este ejercicio es crear una clase Jarra que utilizaremos para “simular” algunas de las acciones que podemos realizar con una jarra.

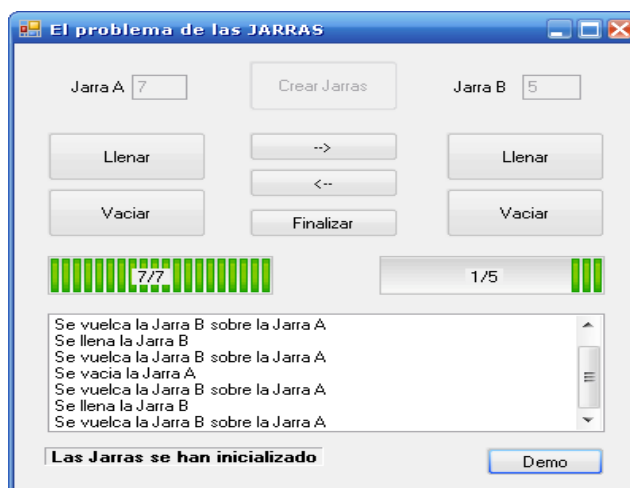
Estas jarras van a poder contener cierta cantidad de líquido. Así, cada jarra tiene una determinada capacidad (en litros) que será la misma durante la existencia de la jarra (proporcionada en el constructor). En un momento determinado una jarra dispondrá de una cantidad de agua (su contenido) que podrá 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. Definir el método toString() que devuelva un String que represente los datos de la jarra.

Jarra
- capacidad : int - contenido : int
+ Jarra(int) + capacidad() : int + cantidad() : int + llenar() + vaciar() + llenarDesde(Jarra) + toString() : String



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.

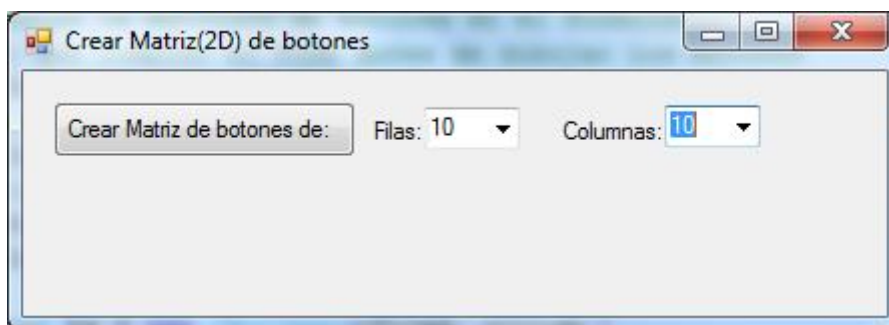
Se desea desarrollar una aplicación gráfica que permita manipular dos jarras. La aplicación tendrá una interfaz gráfica como la de la figura o similar.

19. Realizar una aplicación Windows 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.

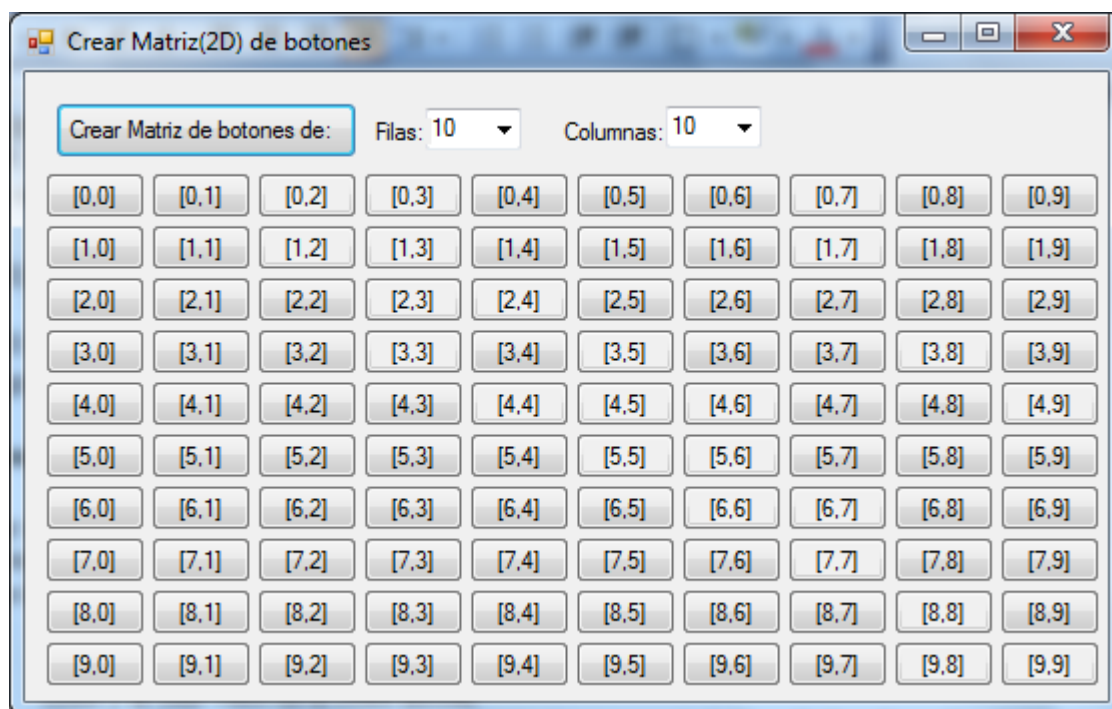
La clase debe cumplir:

- 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:



20. Realizar una aplicación para jugar al juego de tres en raya, dos jugadores o uno con la máquina. Busca en internet información sobre el algoritmo MiniMax que te permite jugar contra la máquina.



21. Haz una aplicación, juego con palabra, que permita acceder a un fichero de texto para obtener una estadística que indique: La palabra mas larga, su longitud y el número de línea en la que está. Igual para la más corta. La palabra que más se repite y la que menos, las palabras palíndromos y otras opciones que consideres oportunas.
22. Haz una aplicación que te permita imprimir cheques bancarios en base a un modelo (imagen) de talonario. Los datos necesarios que debemos recoger son; la cantidad que se introduce en formato numérico, la fecha que por defecto es la actual y el nombre del cobrador que por defecto es 'al portador'.
23. Haz una aplicación Windows para resolver este problema. Un banco realiza préstamos rápidos a sus clientes por un periodo comprendido entre 1 y 24 meses. Para calcular el interés total a pagar, el banco multiplica la cantidad prestada, el número de meses que durará el préstamo y el interés mensual. Si el cliente tiene otras deudas contraídas con el banco, se considera un cliente de riesgo, y se le aplica un recargo extra de un 1 % sobre la suma de la cantidad prestada y el interés total a pagar, y el recargo se distribuye en plazos mensuales iguales a lo largo del periodo que dura el préstamo. Determina un algoritmo que, teniendo en cuenta los valores de la cantidad prestada en euros, el número de meses, el interés mensual (tanto por ciento) y si el cliente es de riesgo o no, calcule la cantidad mensual que debe abonar el cliente.
24. Escribe un programa en Windows que obtenga la rebaja que aplica una compañía de seguros de coches a un asegurado a partir de su edad, sexo y número de siniestros del año anterior, según las siguientes reglas:
  - Si se trata de una mujer mayor de 30 años que no ha tenido ningún accidente, se le aplica un 35 % de descuento;
  - Si el cliente es un hombre mayor de 40 años y no ha tenido ningún accidente, un 20 %;
25. En cualquier otro caso, si el cliente no ha tenido ningún accidente, se beneficia de un descuento del 5 %, independientemente de su sexo y edad.
26. Haz un programa para visualizar imágenes que estén en un dispositivo de almacenamiento. Se debe acceder a una carpeta determinada cargar todas las imágenes y mostrarlas en miniatura al hacer clic sobre una imagen se mostrará en tamaño grande en una ventana distinta.
27. Haz un aplicación que permita la conversión de varios tipos de datos de una clase a mostrar en un interfaz gráfico. La clase contiene un DateTime y las piezas del juego de ajedrez definidas en una enumeración y cada pieza tiene un valor:

```
enum piezasJuego { peon=1, alfil=3, caballo=3, torre=5, reina=10, rey = 20 };
```

El interfaz (imagen inferior izquierda) muestra: fecha corta, el nombre de la pieza y en un ProgressBar su valor. Al lanzar la aplicación la aplicación lee la fecha actual y una pieza al azar que es lo mostrado. Para representar graficamente el valor de la pieza su entero se multiplica por quince que será el valor del gráfico.



28. 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 un nuevo juego, es sólo una idea.

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