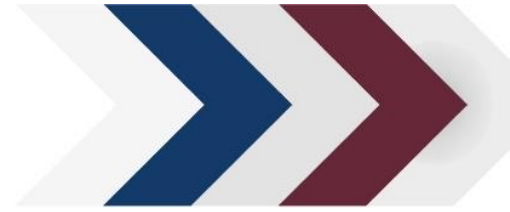


PROGRAMACIÓN I

TP INICIAL LABORATORIOS

UNIDAD 1 INTRODUCCIÓN A LA PROGRAMACIÓN VISUAL

Autor de contenidos:
Nicolás Battaglia



OBJETIVOS

Práctica guiada de laboratorio

ENUNCIADO

EL SIGUIENTE TP SE DIVIDE EN 3 PARTES. EN CADA UNA DE ELLAS ENCONTRARÁ UNA GUÍA PARA REALIZAR LAS PRÁCTICAS DE LABORATORIO

PARTE A

Vamos a comenzar a realizar trabajos en C# y para que no sea tan grande el cambio primero trabajaremos en modo consola que es muy similar a lo que veníamos viendo en programación estructurada

Para ello abrimos el visual studio. Net, elegimos nuevo proyecto, de la pantalla emergente elegimos a la izquierda el lenguaje Visual C#, del centro elegimos aplicación en modo consola.

Abajo del dialogo podemos poner nombre a nuestro proyecto, donde guardarlo y el nombre de la solución

Una solución puede tener más de un proyecto, a nuestro nivel solo haremos un proyecto una solución

Cuando le damos aceptar nos abrirá la pantalla donde trabajaremos

En esta pantalla tenemos el sector del programa.cs específico con su main, también puede verse el explorador de soluciones, la ventana de propiedades, el cuadro de herramientas que utilizaremos más adelante

Escribiremos los siguientes códigos dentro del main en distintos proyectos de distintas soluciones

Una vez finalizado la escritura del código veremos si ejecuta apretando el botón INICIO

Fíjense que a medida que vamos escribiendo código nos aparecen distintos colores de advertencias de posibles errores, azules, rojas, etc. que si nos posicionamos encima de ellos nos dice que error es posible que ocurra, aun aquellos que no lo serán como por ejemplo “**variable que no se ha utilizado**”

Muy bien, pasemos la práctica colocando los códigos dentro de cada main

Ejercicio nro 1

dado dos valores, sumarlos

```
int a;
```



```
float b,c;

a = 10;
b = (float)30.5;
c = (float)(a + b);
Console.WriteLine("{0}+{1}={2}",a,b,c);
Console.ReadKey();
```

Cosas a ver

- Forma de definir variables , similares a dev c++
- (float) forma de transformar ese dato u operación a un valor float
- El console writeline con la secuencia de valores a mostrar
- Console readkey para que pare y podamos ver los resultados

Ejercicio nro 2

dados dos valores indicar si son iguales

```
int a, b;

a = 10;
b = 10;
if (a == b)
{
    Console.WriteLine("son iguales");
    Console.ReadKey();
}
```

Ejercicio nro 3

**ingresar el valor de la hora y el tiempo de trabajo de un
operador , indicar su sueldo**

```
string cadena;
int a, b, c;
Console.WriteLine("ingrese el valor de la hora");
cadena = Console.ReadLine();
a = Convert.ToInt32(cadena);
Console.WriteLine("ingrese el tiempo trabajado");
cadena = Console.ReadLine();
b = Convert.ToInt32(cadena);
c = a * b;
Console.WriteLine(" el sueldo es {0} ",c);
Console.ReadKey();
```

Cosas a ver





- Todos los datos que se ingresan son de tipo string, por lo que hay que utilizar algún tipo de conversión
- Fíjense que la sintaxis del ingreso de datos es al revés que en dev C++
- Para hacerlo más simple primero ingresamos el dato , luego lo convertimos

Ejercicio nro 4 ingresar dos valores, indicar si son iguales y de ser Diferentes indicar cuál es menor

```

        string cadena;
int a, b;
Console.WriteLine("ingrese un valor");
cadena = Console.ReadLine();
a = Convert.ToInt32(cadena);
Console.WriteLine("ingrese otro valor");
cadena = Console.ReadLine();
b = Convert.ToInt32(cadena);

        if (a==b)
{
    Console.WriteLine("son iguales");
}
else
{
    Console.WriteLine("son distintos");
    if (a<b)
    {
        Console.WriteLine("el primero es menor al segundo");
    }
    else
    {
        Console.WriteLine("el segundo es menor al primero");
    }
}
Console.Read();

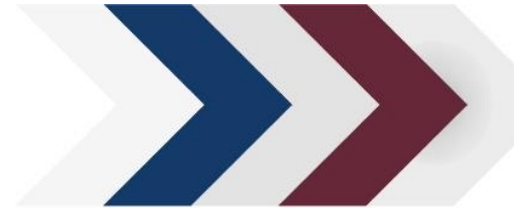
```

Ejercicio nro 5 ingresar 4 valores y sumarlos

```

        string cadena;
int a, tot, i;
tot = 00;
for (i=1;i<=4;i++)
{
    Console.WriteLine("ingrese un valor");
    cadena = Console.ReadLine();
    a = Convert.ToInt32(cadena);
    tot = tot + a;
}

```



```
Console.WriteLine("el total de la suma es {0}", tot);
Console.ReadKey();
```

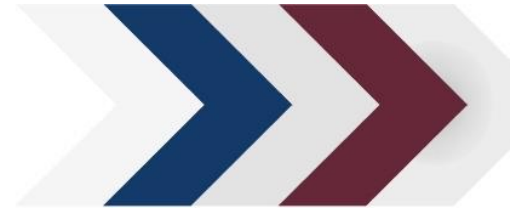
Ejercicio nro 6 ingresar los sueldos de empleados, sumarlos indicar su Valor promedio, finaliza con legajo = 0

```
        string cadena;
int leg, sdo, tot, cont;
float prom;
tot = cont = 0;
prom = 0;
Console.WriteLine("ingrese el legajo");
cadena = Console.ReadLine();
leg = Convert.ToInt32(cadena);
while (leg != 0)
{
    Console.WriteLine("ingrese el sueldo");
    cadena = Console.ReadLine();
    sdo = Convert.ToInt32(cadena);
    tot = tot + sdo;
    cont = cont + 1;
    Console.WriteLine("ingrese el legajo");
    cadena = Console.ReadLine();
    leg = Convert.ToInt32(cadena);
}
    Console.WriteLine("el total pagado es {0}\n", tot);
    Console.ReadKey();
    prom = (float)(tot / cont);
    Console.WriteLine("el sueldo promedio es {0,2:F2}", prom);
    Console.WriteLine("el sueldo promedio es {0}", prom);
    Console.ReadKey();
```

Ejercicio nro 7 ingresar el valor de la hs de cada categoría, guardarlo en Un vector y luego ingresar los empleados de la empresa Hasta legajo = 0, indicar el sueldo de cada empleado y el Total a pagar

```
        string cadena;
//int[] vecval= new int[5];
int[] vecval = Enumerable.Repeat(0, 5).ToArray();
int cat, leg, hs, i, sdo, tot;
tot = 0;
for (i=1; i<=4; i++)
```





```
{
Console.WriteLine("ingrese el valor de la hs de la categoria {0}", i);
cadena = Console.ReadLine();
vecval[i] = Convert.ToInt32(cadena);
}

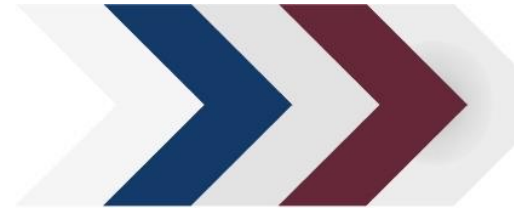
Console.WriteLine("ingrese el nro de legajo");
cadena = Console.ReadLine();
leg = Convert.ToInt32(cadena);
while(leg !=0)
{
Console.WriteLine("ingrese el nro de categoria");
cadena = Console.ReadLine();
cat = Convert.ToInt32(cadena);
Console.WriteLine("ingrese el nro de hs trabajadas");
cadena = Console.ReadLine();
hs = Convert.ToInt32(cadena);
sdo = hs * vecval[cat];
Console.WriteLine("el legajo {0} cobrara {1}", leg, sdo);
tot = tot + sdo;
Console.WriteLine("ingrese el nro de legajo");
cadena = Console.ReadLine();
leg = Convert.ToInt32(cadena);
}
Console.WriteLine("el total a pagar por sueldos es {0}", tot);
Console.ReadKey();
```

Ejercicio nro 8 ingrese el valor de la hs de cada categoría

Indique el sueldo de cada empleado, la cantidad de empleados por categoría

Valor total a pagar en calidad de sueldos

```
string cadena;
//int[]vecval= new int[5];
int[] vecval = Enumerable.Repeat(0, 5).ToArray();
int[] veccant = Enumerable.Repeat(0, 5).ToArray();
int cat, leg, hs, i, sdo, tot;
tot = 0;
for (i = 1; i <= 4; i++)
{
Console.WriteLine("ingrese el valor de la hs de la categoria {0}", i);
cadena = Console.ReadLine();
vecval[i] = Convert.ToInt32(cadena);
}
Console.WriteLine("ingrese el nro de legajo");
cadena = Console.ReadLine();
leg = Convert.ToInt32(cadena);
while (leg != 0)
{
Console.WriteLine("ingrese el nro de categoria");
```



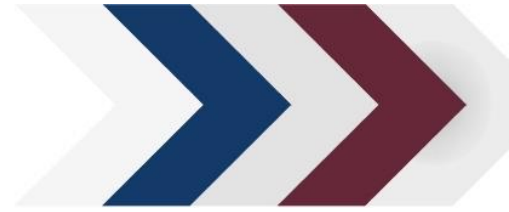
```
cadena = Console.ReadLine();
cat = Convert.ToInt32(cadena);
Console.WriteLine("ingrese el nro de hs trabajadas");
cadena = Console.ReadLine();
hs = Convert.ToInt32(cadena);
sdo = hs * vecval[cat];
Console.WriteLine("el legajo {0} cobrara {1}", leg, sdo);
tot = tot + sdo ;
veccant[cat] = veccant[cat] + 1;
Console.WriteLine("ingrese el nro de legajo");
cadena = Console.ReadLine();
leg = Convert.ToInt32(cadena);
}
for(i=1;i<=4;i++)
{
Console.WriteLine("de la categoria {0} hay {1} empleados", i, veccant[i]);
}
Console.WriteLine("el total a pagar por sueldos es {0}", tot);
Console.ReadKey();
```

Ejercicio nro 9 dados 4 categorías y 4 departamentos ingresar la cantidad de hs trabajas en ellas

```
string cadena;
int[,]matriz = new int[5,5];
//int[] vecval = Enumerable.Repeat(0, 5).ToArray();
//int[] veccant = Enumerable.Repeat(0, 5).ToArray();
int cat, dep, j, i, hs;

for (i = 1; i <= 4; i++)
{
Console.WriteLine("ingrese el valor de la hs de la categoria {0}", i);
cadena = Console.ReadLine();
cat = Convert.ToInt32(cadena);
for (j = 1; j <= 4; j++)
{

Console.WriteLine("ingrese el nro de departamento");
cadena = Console.ReadLine();
dep = Convert.ToInt32(cadena);
Console.WriteLine("ingrese el nro de hs trabajadas");
cadena = Console.ReadLine();
hs = Convert.ToInt32(cadena);
matriz[cat,dep] = matriz[cat,dep] + hs;
```



```
}  
}  
  
    for (i = 1; i <= 4; i++)  
    {  
for (j = 1; j <= 4; j++)  
{  
    Console.WriteLine("categoria {0} depto {1} horas {2}\n", i, j,matriz[i,j]);  
}  
}  
Console.ReadKey();
```

A partir de aquí pueden practicar con algunos ejercicios de programación estructurada a modo de practica antes de pasar al laboratorio numero 1 donde aplicaremos Windows forms en lugar de consola

Ejercicio nro 10: Una pizzería cocina 6 tipos de pizza que son repartidas por medio de sus 4 motos su numerosa clientela en un radio de 20 cuadras a la redonda de su local.

Cuando el pedido es enviado se controla

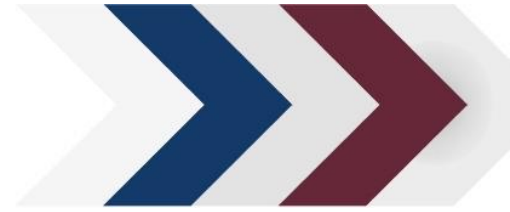
- Nro de ticket
- Código de pizza
- Cantidad
- Nro de moto
- Monto de la venta

Los datos finalizan con el nro de ticket igual a 0.

Se desea saber

- Ticket de mayor valor y que moto llevo ese pedido
- Moto que hizo menos viajes
- Valor promedio de los tickets





- Que porcentaje representa la cantidad de pizzas “A” sobre el total de envíos realizados
- Facturación total del negocio

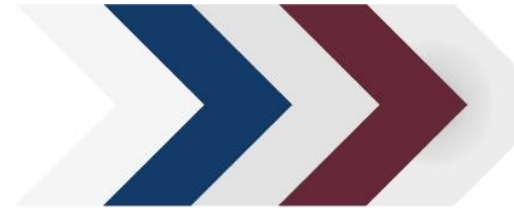
```
static void Main(string[] args)
{
    int[] PRECIO = new int[7] { 10, 20, 30, 4, 5, 6, 7 };
    int MONTOTOTAL = 0, monto;
    int tkmax = 0, montomax = 0, A = 0, cat, moto, motomax, i;
    int tk;
    int cant;
    int[] M = new int[5]; //uso M[0] PARA ACUMULAR MONTO TOTAL
    int motomin;
    int prom = 0;
    int pora = 0;
    Console.WriteLine("INGRESE EL NRO DE TICKET");
    tk = Convert.ToInt32(Console.ReadLine());

    while (tk != 0)
    {
        Console.WriteLine("ingrese la categoria de pizza (de 1 a 6)");
        cat = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("ingrese la cantidad");
        cant = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("ingrese la moto");
        moto = Convert.ToInt32(Console.ReadLine());
        monto = cant * PRECIO[cat];

        if (tkmax == 0)
        {
            tkmax = tk;
            motomax = moto;
            montomax = monto;
        }

        else
        {
            if (monto > tkmax)//obtengo monto maximo
            {
                tkmax = tk;
                motomax = moto;
                montomax = monto;
            }
        }
        M[moto]++; //acumulo cantidad de viajes por moto
        MONTOTOTAL = MONTOTOTAL + monto; //acumulo monto
        M[0]++; //ACUMULO CANTIDAD DE VIAJES TOTALES

        if (cat == 1)
        {
            A++;
        }
    }
}
```



```

Console.WriteLine("INGRESE EL NRO DE TICKET");
tk = Convert.ToInt32(Console.ReadLine());    //VUELVO a pedir ingreso de ticket para
que ingrese el proximo

    }//fin ingreso

    prom = MONTOTOTAL / M[0];    //calculo el promedio
    pora = A / M[0];
    motomin = 1; // declaro la moto minima
    for (i = 1; i <= 4; i++)//calculo moto menos viajes
    {
if (M[i] < M[motomin])
{
    motomin = i;
}
    }    //fin calculo moto menos viajes
    // comienzo a mostrar
    Console.WriteLine("el ticket {0} de valor: {1} es el de mayor importe", tkmax,
montomax);
    Console.WriteLine("la moto {0} es la que hizo menos viajes", motomin);
    Console.WriteLine("el valor promedio de los viajes es {0} ", prom);
    Console.WriteLine("el porcentaje de pedidos de pizzas categoria A es {0}",
pora);
    Console.WriteLine("la facturacion total del negocio es {0}", MONTOTOTAL);
    Console.ReadKey();

}

```



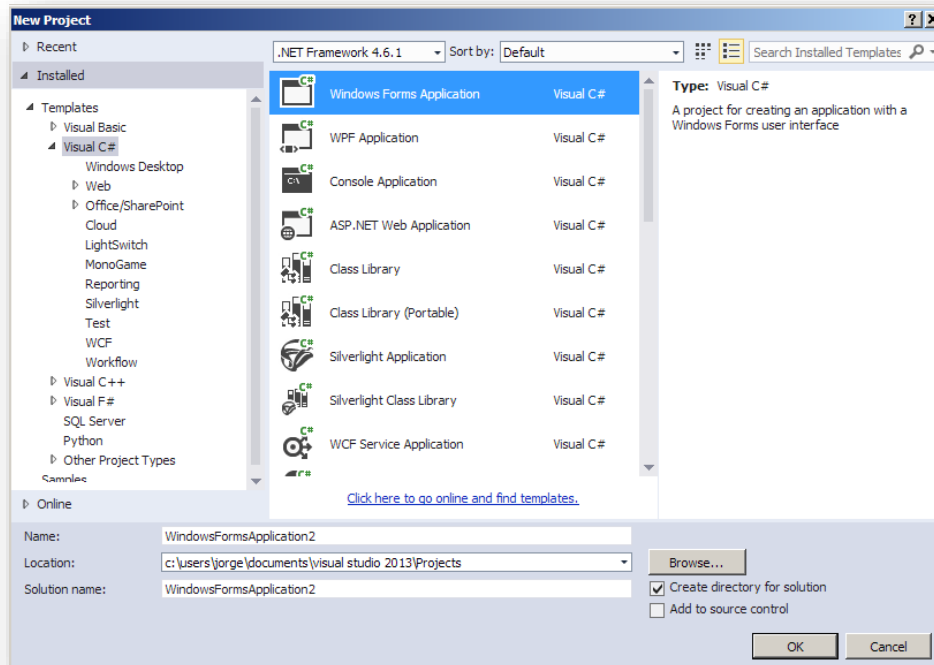


PARTE B

Ingremos al **Visual Studio**

Elijamos
crear
proyecto

En la
siguiente
pantalla
C#,
elegimos
**Windows
Forms**



Application y coloquemos un nombre al proyecto, indicando la ubicación donde queremos guardar dicho proyecto.

Al dar aceptar entraremos en el IDE de .Net, donde debemos tener a la vista o en solapas a cada lado de la pantalla lo siguiente:

- Solapa de Cuadro de herramientas
- Solapa de Explorador de proyectos
- Solapa de Propiedades

De no estar las agregaremos desde el menú Ver.

Tendremos también a la vista el primer **Form** donde trabajar, iremos al cuadro de herramientas y seleccionaremos los siguientes objetos que colocaremos dentro de este formulario **3 textbox, 3 label y 5 buttons**

Para esto podemos seleccionarlo y arrastrarlo al form o hacer doble click sobre el y automáticamente se agregara en el form.

Realizaremos el diseño según la pantalla siguiente y cambiaremos algunas propiedades de los controles



Las propiedades a modificar serán

| Controles | text | Nombre |
|-----------|--------------------|----------------|
| Form1 | Calculadora simple | frmcalculadora |
| Label1 | 1er operador | Lblop1 |
| Label2 | 2do operador | Lblop2 |
| Label3 | Resultado | Lblrta |
| Textbox1 | | Txtp1 |
| Textbox2 | | Txtp2 |
| Textbox3 | | Txtrta |
| Button1 | Suma | Btnsuma |
| Button2 | Resta | Btnresta |
| Button3 | Producto | Btnprod |
| Button4 | Cociente | Btncoc |
| Button5 | Salir | Btnsalir |

Podremos modificar algunas otras propiedades como

BackColor, **size** (siempre y cuando locked sea false), **autosize** , etc.

(Jueguen e investiguen, de paso les va a ir sirviendo para el TP nro 2)





Para igualar los tamaños de los controles los seleccionaremos, iremos a **formato, igualar tamaños** y los dejara del tamaño del primero seleccionado (fijense que los puntos alrededor de este objeto están blancos mientras en los otros objetos seleccionados están negros); allí mismo podemos **alinear, dar espacios, centrar**, etcétera dichos controles.

Muy bien, pasemos al código

En cada botón escribiremos

En el evento Click del botón Resta escribimos

```
txtrta.Text = Convert.ToString(Convert.ToInt32(txttop1.Text) - Convert.ToInt32(txttop2.Text));
```

En el evento Click del botón Producto escribimos

```
txtrta.Text = (Int32.Parse(txttop1.Text) * Int32.Parse(txttop2.Text)).ToString();
```

En el evento Click del botón Cociente escribimos

```
txtrta.Text = (Int32.Parse(txttop1.Text) / Int32.Parse(txttop2.Text)).ToString();
```

En el evento Click del botón Suma escribimos

```
txtrta.Text = (Int32.Parse(txttop1.Text) + Int32.Parse(txttop2.Text)).ToString();
```

Es importante recordar que hay distintas formas de conversión, implícitas y explícitas. Dado que los tipos utilizados en el formulario no son de la misma familia como entero y doble, es necesario utilizar una conversión explícita.





Para ello recurrimos a `Convert.XX` o bien escribimos el tipo al que queremos convertir acompañado de `".Parse"`.

En el evento Click del botón Cociente escribimos lo siguiente

```
int num = Int32.Parse(txttop1.Text);  
int den = Int32.Parse(txttop2.Text);  
if (den != 0)  
{  
    txttrta.Text = (Int32.Parse(txttop1.Text) / Int32.Parse(txttop2.Text)).ToString();  
}
```

En el evento Click del botón Limpiar escribimos

```
txttop1.Text = "";  
txttop2.Text = "";  
txttrta.Text = "";  
this.txttop1.Focus();
```

En el evento Click del botón Salir escribimos

```
Close();
```

A continuación veremos algunos temas sueltos que en general a esta altura de la cursada los alumnos comienzan a preguntar





Tengan en cuenta que no podemos ver todo de una vez y que cada uno de ustedes tiene sus propias inquietudes

Como hacer

Para modificar el tamaño de algunos controles

Por defecto algunos controles solo tienen activados algunos de sus manejadores, mientras que los otros están atenuados.

Esto se debe a la propiedad autosize que esta por defecto en True.

Para modificar el tamaño se debe modificar la propiedad size, width y height.

Para modificar el texto dentro de un label

Se modifica la propiedad text

Como guardar el proyecto

Es conveniente guardar periódicamente el trabajo, para eso ir a archivos, guardar todo, si se agregó algo nuevo C# pedirá un nombre para esos elementos nuevos y su ubicación.

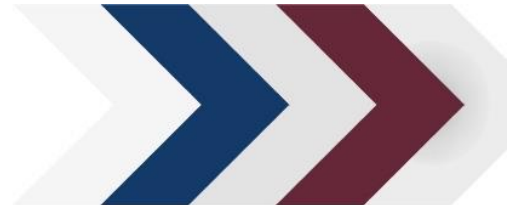
Que es un evento y como funciona o se puede modificar

La mayoría de los controles responden ante distintos eventos, cada uno de ellos tiene un evento predeterminado, que es el que suele ocurrir más frecuentemente, el programador puede modificar esto al programar.

Para declarar dos variables del mismo tipo de una vez

C# posibilita la siguiente declaración de dos variables de tipo entero

```
int var1, var2;
```



Para poner un comentario

El compilador ignora estas líneas.

```
// comentario de una línea

/*
comentario de más de una línea
necesario para aclarar
que hace el programa


*/
```

Que son las soluciones, proyectos, su diferencia

Las soluciones contienen elementos que necesita para crear la aplicación. Una solución incluye uno o más proyectos, además de los archivos y metadatos que ayudan a definir la solución como un todo. Visual Studio genera una solución automáticamente cuando se crea un nuevo proyecto. Visual Studio almacena la definición de una solución en dos archivos: .sln y .suo. El archivo de definición de soluciones (.sln) almacena los metadatos que definen la solución. ([https://msdn.microsoft.com/es-es/library/b142f8e7\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/b142f8e7(v=vs.110).aspx))

El proyecto contiene todo el material necesario para la aplicación: los archivos de código fuente, archivos de recursos, tales como iconos, referencias a archivos externos de los que depende la aplicación, y datos de configuración, tales como las opciones del compilador. Cuando se genera un proyecto, Visual C# invoca al compilador de C# y a otras herramientas internas para crear un ensamblado ejecutable con los archivos del proyecto. ([https://msdn.microsoft.com/es-es/library/ms173077\(v=vs.90\).aspx](https://msdn.microsoft.com/es-es/library/ms173077(v=vs.90).aspx))

Fijar u ocultar ventanas del IDE

Para fijar u ocultar deberá hacer clic sobre el pin correspondiente de esa ventana .

Para buscar código escrito en mi proyecto

| | |
|-----------|---------------------|
| Ctrl. + F | inicia una búsqueda |
| F3 | repite la búsqueda |



Ctrl. + H busca y reemplaza

Variables

Una variable representa un valor numérico o de cadena o un objeto de una clase. El valor que la variable almacena puede cambiar, pero el nombre sigue siendo el mismo. Una variable es un tipo de *campo*. El código siguiente es un ejemplo sencillo de cómo declarar una variable de entero, asignarle un valor y, a continuación, asignarle un nuevo valor.

En C#, las variables se declaran con un tipo de datos y una etiqueta concretos. Si hasta ahora sólo ha utilizado lenguajes con tipos definidos de forma imprecisa como JScript, estará acostumbrado a emplear el mismo tipo "var" para todas las variables, pero en C# tiene que especificar si la variable es de tipo **int**, **float**, **byte**, **short** u otro cualquiera entre más de 20 tipos de datos diferentes. El tipo especifica, entre otras cosas, la cantidad de memoria exacta que se debe asignar para almacenar el valor cuando la aplicación se ejecuta. El lenguaje C# fuerza ciertas reglas al convertir una variable de un tipo en otro.

| Tipo de datos | Intervalo |
|---------------|---|
| byte | 0 .. 255 |
| sbyte | -128 .. 127 |
| short | -32,768 .. 32,767 |
| ushort | 0 .. 65,535 |
| int | -2,147,483,648 .. 2,147,483,647 |
| uint | 0 .. 4,294,967,295 |
| long | -9,223,372,036,854,775,808 .. 9,223,372,036,854,775,807 |
| ulong | 0 .. 18,446,744,073,709,551,615 |





| | |
|----------------|--|
| float | -3.402823e38 .. 3.402823e38 |
| double | -1.79769313486232e308 .. 1.79769313486232e308 |
| decimal | -79228162514264337593543950335 .. 79228162514264337593543950335 |
| char | Un carácter Unicode. |
| string | Una cadena de caracteres Unicode. |
| bool | True o False. |
| object | Un objeto. |

([https://msdn.microsoft.com/es-ar/library/wew5ytx4\(v=vs.90\).aspx](https://msdn.microsoft.com/es-ar/library/wew5ytx4(v=vs.90).aspx))

Nombrar una variable

- El nombre debe empezar por una letra
- El resto del nombre puede contener letras, numeros o caracteres subrayados , no se permiten espacios puntos u otros signos de puntuación
- El nombre debera ser unico dentro del alcance de la variable
- El nombre no puede ser una palabra reservada de C#

Como regla conveniente podemos tener

Si la variable es de un tipo string el prefijo sera str como ejemplo strnombre

| | | |
|--------------|-----|------------|
| Integer | int | intnumero |
| Long integer | lng | lngfortuna |
| Double | dbl | dblradio |
| Booleano | b | bflag |

Normalmente el valor predeterminado para las variables es 0





El tipo de dato universal, es el object

Para cambiar en un formulario o form

| Accion | Propiedad |
|-----------------------|---------------------------|
| Los bordes | Formborderstyle |
| Poner o sacar botones | Maximizebox o minimizebox |
| Tamaño | Size |
| Posición | Location |
| Posición inicial | StartPosition |
| Colores | BackColor |

Ingresar una fecha tipo calendario

1. Pegue en el form un control datetimepicker
2. Pegue en el form un control monthcalendar



Cual es la diferencia?

Para activar un control de tiempo

A veces, conviene crear un procedimiento que se ejecuta a intervalos de tiempo específicos hasta que finaliza un bucle o que se ejecuta cuando ha transcurrido un intervalo de tiempo establecido. **El componente Timer** hace posible este procedimiento.



Pegue en el form un control timer. Este control sera invisible durante la ejecución.

El objeto Timer posee una propiedad llamada **Interval** la cual nos permite definir el tiempo que una acción se ejecutará periódicamente.

El objeto Timer posee un método **Tick** en el cual se debe escribir el código que deseamos que se ejecute de acuerdo a la frecuencia definida en Interval.

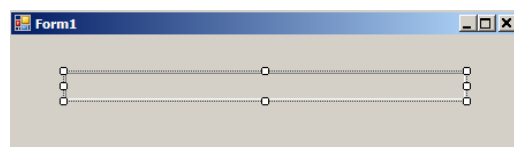
([https://msdn.microsoft.com/es-es/library/3tszykws\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/3tszykws(v=vs.110).aspx))

Para activar una barra de control de progreso

Colocar un control **progressbar** en el formulario.

Se deben fijar las propiedades:

- ❖ Máximum : determina el valor máximo a representar
- ❖ Minimum: determina el valor mínimo a representar
- ❖ Value: es el valor que se muestra en la barra de progreso
- ❖ Style : determina la forma en que se muestra la actividad en segundo plano (investigue Marque)
- ❖ Step : es el valor que se establece para definir pasos de tamaño fijo

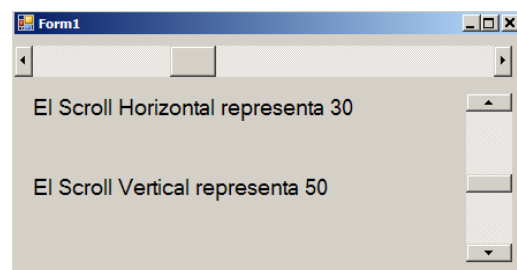


Value se puede definir en forma aleatoria de acuerdo a los cálculos que el programador desee mostrar.

Para colocar una barra de scroll o desplazamiento

Colocar un control Hscrollbar o Vscrollbar en el formulario y controlar las propiedades maximun, minimun y value del mismo

Para ver en un label el valor de la posición donde esta el scroll coloco en el codigo
label1.text = Hscrollbar.value;



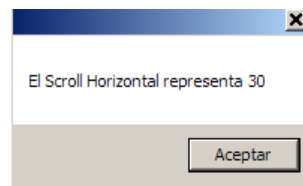
Para ver mensajes de texto ante una situación dada



Utilice el objeto MessageBox.

El método Show expone los parámetros asignados.

El objeto MessageBox tiene 21 sobrecargas que permiten gran flexibilidad a la hora de mostrar resultados.



```
MessageBox.show("mensaje", "titulo del mensaje");
```

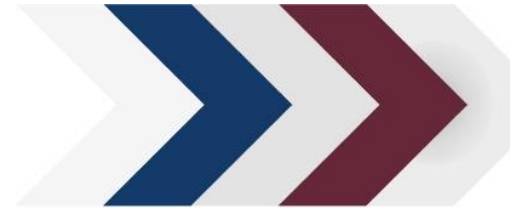
```
MessageBox.show("mensaje", "titulo del mensaje", messageboxbuttons.yesno);
```

```
MessageBox.show("mensaje", "titulo del mensaje", messageboxbuttons.yesno, asterik);
```

¿Que diferencia existe entre ellos?

En el evento Load del formulario incluir el siguiente código:

```
hScrollBar1.Value = 30;  
vScrollBar1.Value = 50;  
label1.Text = "El Scroll Horizontal representa " + (this.hScrollBar1.Value).ToString();  
label2.Text = "El Scroll Vertical representa " + (this.vScrollBar1.Value).ToString();  
MessageBox.Show(label1.Text);
```



PARTE C

Objetivo:

Realizar ejercitación en C# con elementos conocidos, como lo son los condicionales IF y SWITCH..

Así mismo se incorporaran los objetos radiobutton, groupbox y checkbox para la implementación de estos temas

Nota importante

En este laboratorio tendremos varios formularios, es por eso que para agregar un formulario a nuestra solución deberemos ir al explorador de proyecto, pararnos sobre el nombre del proyecto, botón derecho, agregar, nuevo elemento, Windows form.

En el caso de la pantalla de bienvenida elegir a esta en lugar de Windows forms menú

Ingresa a C#, cree un proyecto y realice el siguiente formulario

Agregue los buttons y radiobuttons de acuerdo al siguiente diseño

Las propiedades que modificaremos en los controles serán

| control | text | Name |
|---------|----------------------------|----------------|
| Form | Calculadora con radioboton | frmradiobutton |
| Button1 | Limpiar | btnLimpiar |
| Button2 | Con if | Btnif |
| Button3 | Con Switch | btnswitch |
| Button4 | Salir | Btnsalir |
| Label1 | 1er operador | Lbl1 |
| Label2 | 2do operador | Lbl2 |



| Label3 | Resultado | Lbl3 |
|--------------|-------------|-----------|
| Textbox1 | | Txt1 |
| Textbox2 | | Txt2 |
| Textbox3 | | TxtRta |
| GroupBox | Operaciones | GroupBox1 |
| Radiobutton1 | Suma | Optsuma |
| Radiobutton2 | Resta | Optresta |
| Radiobutton3 | Producto | Optprod |
| Radiobutton4 | cociente | OptCoc |

En el button **Limpiar** colocaremos el siguiente código

ACLARAR PORQUE USAMOS THIS

```
private void btnLimpiar_Click(object sender, EventArgs e)
{
    this.Txt1.Text = null;
    this.Txt2.Text = null;
    this.TxtRta.Text = null;
}
```

En el button **con if** colocaremos el siguiente código

```
private void btnIf_Click(object sender, EventArgs e)
{
    if (this.Optresta.Checked)
    {
```





```
        this.TxtRta.Text = (Int32.Parse(Txt1.Text) -
Int32.Parse(Txt2.Text)).ToString();
    }

    if (this.OptSuma.Checked)
    {
        this.TxtRta.Text = (Int32.Parse(Txt1.Text) -
Int32.Parse(this.Txt2.Text)).ToString();
    }

    if (OptCoc.Checked)
    {
        if (Int32.Parse(this.Txt2.Text) != 0)
        {
            this.TxtRta.Text = (Int32.Parse(this.Txt1.Text) /
Int32.Parse(this.Txt2.Text)).ToString();
        }
    }

    if (OptProd.Checked)
    {
        this.TxtRta.Text = (Convert.ToInt32(this.Txt1.Text) *
Convert.ToInt32(this.Txt2.Text)).ToString();
    }

    if (this.OptSuma.Checked)
    {
        this.TxtRta.Text = (Convert.ToInt32(this.Txt1.Text) +
Convert.ToInt32(this.Txt2.Text)).ToString();
    }
}
```





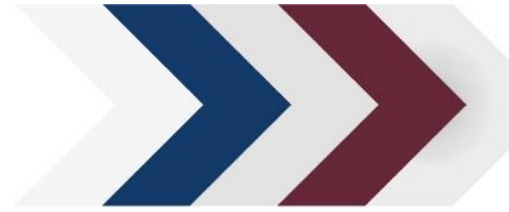
En el botón **con Switch** colocaremos el siguiente código

```
private void btnswitch_Click(object sender, EventArgs e)
{
    bool mayor_cero = true;
    int opcion = 0;
    if (this.Txt1.Text != null && this.Txt2.Text != null)
    {
        if (Int32.Parse(this.Txt2.Text) != 0)
        {
            mayor_cero = true;
        }
        else
        {
            mayor_cero = false;
        }

        if (this.OptSuma.Checked) opcion = 1;
        if (this.Optresta.Checked) opcion = 2;
        if (this.OptProd.Checked) opcion = 3;
        if (this.OptCoc.Checked)
        {
            if (mayor_cero) opcion = 4;
        }

        switch (opcion)
        {
            case 1:
                this.TxtRta.Text = (Int32.Parse(Txt1.Text) +
Int32.Parse(this.Txt2.Text)).ToString();
```





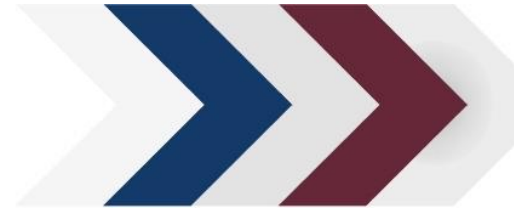
```

        break;
    case 2:
        this.TxtRta.Text = (Int32.Parse(Txt1.Text) -
Int32.Parse(Txt2.Text)).ToString();
        break;
    case 3:
        this.TxtRta.Text = (Int32.Parse(this.Txt1.Text) *
Int32.Parse(this.Txt2.Text)).ToString();
        break;
    case 4:
        this.TxtRta.Text = (Int32.Parse(this.Txt1.Text) /
Int32.Parse(this.Txt2.Text)).ToString();
        break;
    default:
        string mensaje = "La operación no se puede realizar";
        string titulo = "Importante";
        MessageBoxButtons botones = MessageBoxButtons.YesNo;
        DialogResult resultado;
        resultado = MessageBox.Show(mensaje, titulo, botones);
        if (resultado == System.Windows.Forms.DialogResult.Yes)
        { limpiar(); }
        break;
    }
}
}
}

```

A partir de aquí el nombre de los objetos deberan ser deducidos por uds del codigo que sigue al diseño de los mismos

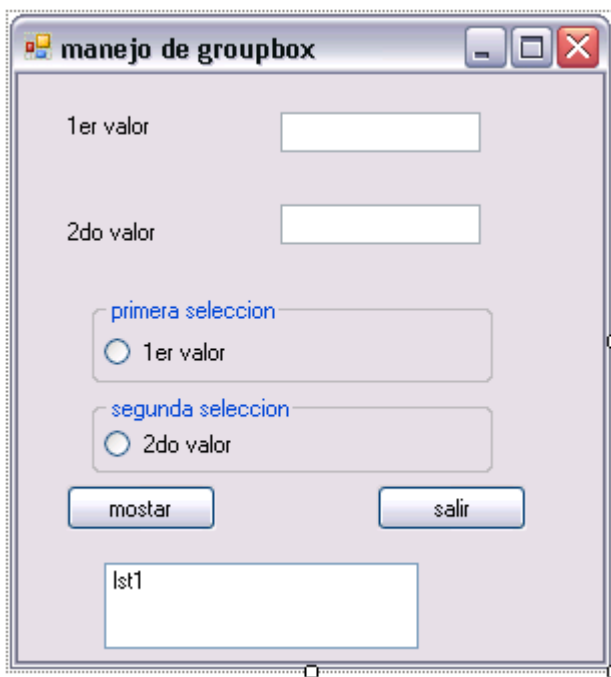
Muy bien. Finalizado este formulario agregaremos otro a nuestro proyecto con el siguiente diseño.



Los dos radiobutton se encuentran insertados dentro de respectivos groupbox.

Así podremos tenerlos simultáneamente seleccionados y el resultado no lo mostraremos en un textbox sino en un listbox.

Atención !!! El objeto lst1 no es un textbox agrandado, sino un objeto listbox seleccionado del cuadro de herramientas.



Dentro del button mostrar codificaremos lo siguiente

```
if (opt1.Checked == true)
{
```



```
lst1.Items.Add(textBox1.Text);  
opt1.Checked = false;  
}  
if (opt2.Checked == true)  
{  
    lst1.Items.Add(textBox2.Text);  
    opt2.Checked = false;  
}
```

Dentro del button Limpiar codificaremos lo siguiente

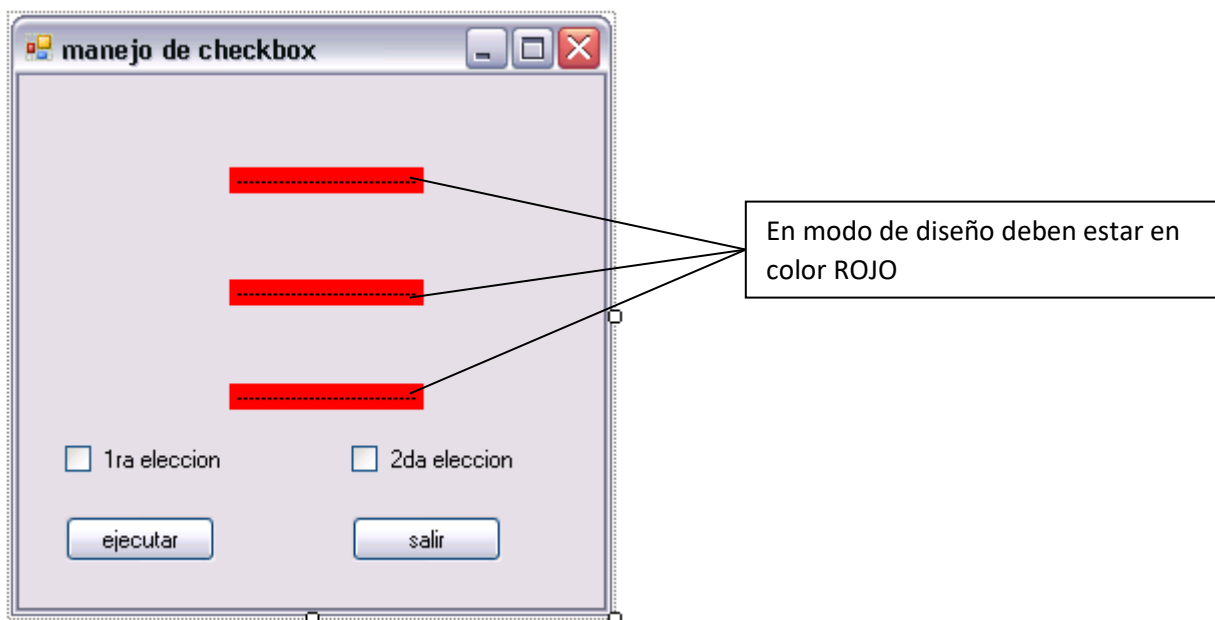
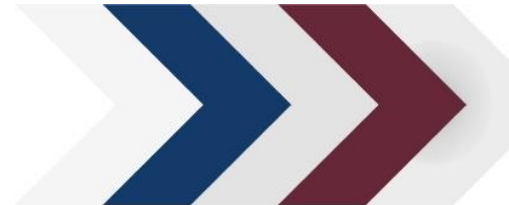
```
lst1.Items.Clear();
```

quando el codigo del boton Salir

```
close();
```

por ultimo generaremos otro formulario donde trabajaremos con el control checkbox , este nos permitira sin necesidad de tener los groupbox , seleccionar varias opciones a la vez

El diseño de este nuevo formulario sera el siguiente



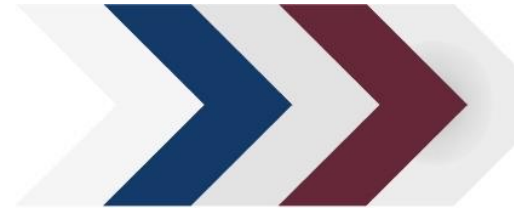
los controles y propiedades a modificar seran

| control | text | name | backcolor |
|-----------|--------------|-------------|-----------|
| Button1 | Ejecutar | btnejecutar | |
| Button2 | Salir | btnsalir | |
| Checkbox1 | 1ra elección | Chk1 | |
| Checkbox2 | 2da elección | Chk2 | |
| Label1 | | Lbl1 | Rojo |
| Label2 | | Lbl2 | Rojo |
| Label3 | | Lbl3 | Rojo |

Escribamos ahora en el botón ejecutar lo siguiente

```
if ((this.Chk1.Checked == true) & (this.Chk2.Checked == false))
{
```

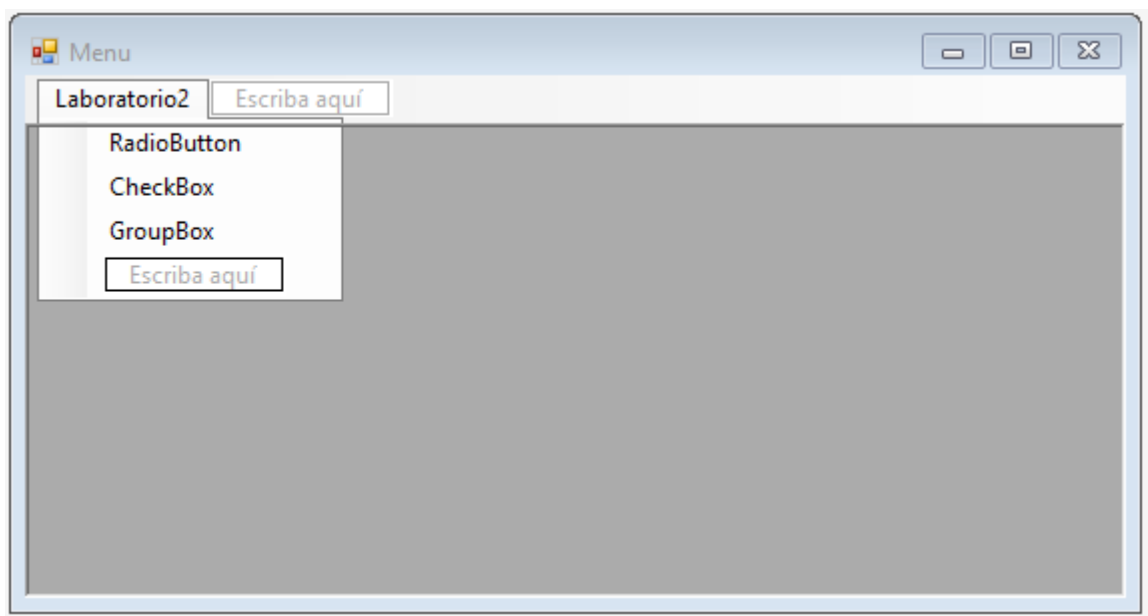
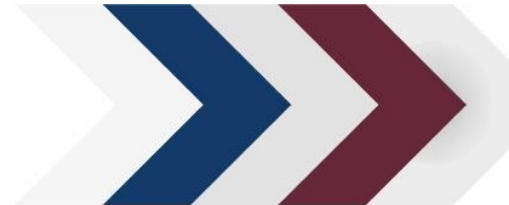




```
Lbl1.BackColor = Color.Aquamarine;
Lbl2.BackColor = Color.Beige;
Lbl3.BackColor = Color.Black;
}
if ((Chk1.Checked == false) & (Chk2.Checked == true))
{
    Lbl1.BackColor = Color.Green;
    Lbl2.BackColor = Color.LightPink;
    Lbl3.BackColor = Color.Linen;
}
if ((Chk1.Checked == true) & (Chk2.Checked == true))
{
    Lbl1.BackColor = Color.Blue;
    Lbl2.BackColor = Color.White;
    Lbl3.BackColor = Color.Blue;
}
```

Por ultimo veremos de unir todos estos formularios con un menú

Para ellos incorporaremos un formulario MDI (contenedor) al que llamaremos Menu, para ello enen propiedades **IsMdiContainer = true**, y al que le agregaremos un control **menustrip** donde codificaremos en cada uno de los ítems del menustrip el nombre del formulario que se desea mostrar



En cada submenu se escribe el siguiente código

```
//Creo el formulario que deseo mostrar
frmRadioButton ofrmRadioButton = new frmRadioButton();

//Digo que es Hijo del MDI
ofrmRadioButton.MdiParent = this;

//Muestro el form
ofrmRadioButton.Show();
```

En forma genérica

```
//Creo el formulario que deseo mostrar
Nombre_del_formulario_real Nombre_elegido = new Nombre_del_formulario_real();

//Digo que es Hijo del MDI
Nombre_elegido.MdiParent = this;

//Muestro el form
Nombre_elegido.Show();
```



Si quiero armar una pantalla de Bienvenida en el MDI o cualquier form.

Voy a la propiedad backgroundImage y selecciono la imagen que deseo

