



# Índice

| MÓDULO I: INTRODUCCIÓN A .NET   | 3  |
|---------------------------------|----|
| Ejercicio 1                     | 3  |
| EJERCICIO 2                     |    |
| EJERCICIO 3                     | 4  |
| MÓDULO II: INTRODUCCIÓN A C#    | 7  |
| Ejercicio 1                     | 7  |
| EJERCICIO 2                     | 8  |
| EJERCICIO 3                     | 9  |
| EJERCICIO 4                     | 10 |
| EJERCICIO 5                     | 11 |
| MÓDULO III: DIAGRAMAS DE CLASES | 13 |
| Ejercicio 1                     | 13 |
| EJERCICIO 2                     | 15 |
| MÓDULO IV: POO CON C#           | 17 |
| EJERCICIO 1                     | 17 |
| EJERCICIO 2                     |    |
| EJERCICIO 3                     | 20 |
| EJERCICIO 4                     | 21 |
| EJERCICIO 5                     | 23 |
| MÓDULO V: XHTML Y SCRIPTS       | 24 |
| Ejercicio 1                     | 24 |
| EJERCICIO 2                     | 24 |
| EJERCICIO 3                     | 25 |
| Ejercicio 4                     | 26 |





# Módulo I: Introducción a .Net

# Ejercicio 1

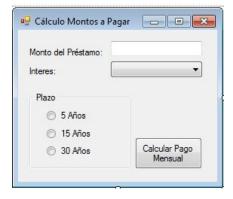
En este ejercicio se trabajará con proyectos de .Net. También se analizarán las principales herramientas del ambiente de desarrollo.

#### Pasos:

- 1. Abrir Visual Studio 2008.
- 2. En el menú File, hacer click en New, y luego en Project.
- 3. Aparecerá el cuadro de diálogo New Project.
- 4. Para elegir el tipo de proyecto, expandir Visual C# y hacer click en Windows, seleccionar la plantilla Aplicación Windows Forms.
- 5. Especificar un nombre y una ubicación para el proyecto y presionar **OK**. Visual Studio creará el proyecto.
- 6. Antes de comenzar, veamos básicamente qué contiene el entorno de desarrollo.
  - a. A la derecha de la aplicación se encuentra la ventana Solution Explorer. Ésta nos permite ver y administrar los archivos, referencias, configuración y recursos del proyecto.
  - b. En la parte superior izquierda está la ventana **Toolbox**. Contiene los controles que se pueden utilizar en el formulario, como botones, cuadros de texto, etc.
  - c. La ventana **Properties** de la parte inferior derecha permite establecer las propiedades y los eventos de los controles.
  - d. En la parte central se encuentra el editor de código.
- Agregar una biblioteca de clases a la solución. En el menú File, hacer click en Add y luego en New Project. Expandir Visual C# y hacer click en Windows, seleccionar la plantilla Class Library.
- 8. En la ventana Solution Explorer, hacer click con el botón secundario del mouse sobre el nodo References perteneciente al proyecto del formulario y seleccionar Add Reference. En el cuadro de diálogo, hacer click sobre la solapa Projects y seleccionar el proyecto de tipo Biblioteca de Clases, presionar OK.
- 9. La referencia que agregó aparecerá en el nodo References

# Ejercicio 2

Sobre el proyecto Windows creado en el Ejercicio 1 diseñar el siguiente formulario y escribir el evento click de botón de manera que muestre un mensaje



#### Pasos:





- 1. Poner los controles sobre el formulario arrastrando los mismos desde la **ToolBox** y acomodarlos de manera que el formulario quede similar al de la imagen.
- 2. Configurar las siguientes propiedades del formulario y controles.

Formulario:

Name: frmPrestamos

Text: Cálculo de Montos a Pagar

Monto del Préstamo:

Label: Name: IblMonto

Text: Monto del Préstamo:

Textbox: Name: txtMonto TextAlign: Right

Interes:

Label: Name: IblInteres
Text: Interes:

ComboBox: Name: cboInteres

Items: 10, 15, 20 y 24

DropDownStyle: DropDownList

Plazo:

3 RadioButton dentro de un GroupBox

GroupBox:

Name: grbPlazo Text: Plazo

RadioButtons:

Name: rdb5Años Text: 5 Años Name: rdb15Años Text: 15 Años Name: rdb30Años Text: 30 Años

Calcular Pago Mensual:

Button: Name: btnPagoMensual Text: Calcular Pago Mensual

- 3. Hacer doble click sobre el botón de manera de generar el atajador del evento Click
- Escribir un MessageBox mostrando un mensaje que diga "Probando la Aplicación". El mensaje debe tener un título que diga "Aplicación de Prueba", mostrar solo un botón de Ok y mostrar un icono informativo.
- 5. Probar el ejercicio

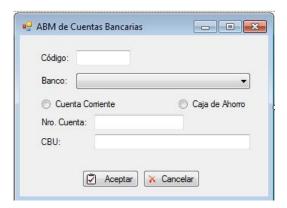
#### Solución:

```
private void btnConvertir_Click(object sender, EventArgs e)
{
    MessageBox.Show("Probando la Aplicación", "Aplicación de
Prueba", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
```

# Ejercicio 3

En este ejercicio diseñara un nuevo formulario. Las imágenes necesarias se encuentran en la carpeta de la solución del ejercicio. Puede completar el ejercicio usando cualquier imagen.





#### Pasos:

- 1. Abrir Visual Studio 2008.
- 2. En el menú File, hacer click en New, y luego en Project.
- 3. Aparecerá el cuadro de diálogo New Project.
- 4. Para elegir el tipo de proyecto, expandir **Visual C#** y hacer click en **Windows**, seleccionar la plantilla **Aplicación Windows Forms**.
- 5. Especificar un nombre y una ubicación para el proyecto y presionar **OK**. Visual Studio creará el proyecto.
- 6. Poner los controles sobre el formulario arrastrando los mismos desde la **ToolBox** y acomodarlos de manera que el formulario quede similar al de la imagen.
- 7. Configurar las siguientes propiedades del formulario y controles.

Formulario:

Name: frmABMBancos

Text: ABM de Cuentas Bancarias

Código:

Label: Name: IblCodigo

Text: Código: Textbox: Name: txtCódigo

TextAlign: Right

MaxLength: 5

Banco:

Label: Name: IblBanco

Text: Banco:

ComboBox: Name: cboBancos

Items: Santander, Galicia, Francés DropDownStyle: DropDownList

Plazo:

2 RadioButton RadioButtons:

Name: rdbCtaCte Text: Cuenta Corriente Name: rdbCajaAhorro Text: Caja de Ahorro

Nro.Cuenta:

Label: Name: IblNumero

Text: Nro. Cuenta:

Textbox: Name: txtNumero

MaxLength: 15



SISTEMAS Av de Mayo 676 Cap Fed

CBU:

Label: Name: IblCBU

Text: CBU: Textbox: Name: txtCBU

MaxLength: 30

Aceptar / Cancelar

Button: Name: btnAceptar Text: Aceptar

TextAlign: Centrado – Derecha

Image: Aceptar.ico

ImageAlign: Centrado - Izquierda

Button: Name: btnCancelar Text: Cancelar

TextAlign: Centrado – Derecha

Image: Cancelar.ico

ImageAlign: Centrado - Izquierda

- 8. Hacer doble click sobre el botón Aceptar de manera de generar el atajador del evento **Click**
- 9. Escribir un **MessageBox** mostrando un mensaje que muestre el código y el banco ingresados.
- 10. Probar el ejercicio

#### Solución:

```
private void btnAceptar_Click(object sender, EventArgs e)
{
    MessageBox.Show(txtCodigo.Text + " - " + cboBancos.Text);
}
```





# Módulo II: Introducción a C#

#### Ejercicio 1

En este ejercicio, se trabajará con variables y las herramientas de **debug**.

#### Pasos:

- 1. Abrir una nueva instancia de Visual Studio 2008.
- En el menú File, hacer click en New, y luego en Project o hacer click en Create Project de la ventana Start Page.
- 3. Aparecerá el cuadro de diálogo New Project.
- 4. Para elegir el tipo de proyecto, expandir el nodo **Visual C#** y hacer click en **Windows**, seleccionar la plantilla **Windows Forms Application**.
- Especificar un nombre y una ubicación para el proyecto y presionar OK. Visual Studio creará el proyecto.
- 6. Para agregar una etiqueta al formulario, desde la ventana **Toolbox** arrastar tres controles **Label** al formulario. Acomodarlos.
- 7. Para agregar el botón, desde la ventana **Toolbox** arrastar un control **Button** al formulario. Presionar doble click sobre el button para crear el código del evento **Click**.
- 8. Declarar, inicializar y mostrar en la primera etiqueta una variable con el valor **Juan Pérez**.
- 9. Declarar, inicializar y mostrar en la segunda etiqueta una variable con el valor **135.20**. Se asume que es una variable de tipo **decimal**.
- Crear una enumeración llamada DiasSemana, agregar los días de la semana a la enumeración.
- Declarar una variable Dia, asignar un valor a DiasSemana. Martes y mostrarlo en la etiqueta.
- 12. Usar las herramientas de **debug** para ver el código paso a paso, y examinar los valores de las variables, usando la ventana **Autos**.
- 13. Probar el ejercicio

#### Solución:

```
namespace Ejerciciol
{
    enum DiasSemana
    {
        Lunes,
        Martes,
        Miercoles,
        Jueves,
        Viernes,
        Sabado,
        Domingo
    }

    public partial class Form1 : Form
    {
        private void button1_Click(object sender, EventArgs e)
        {
            string Nombre = "Juan Pérez";
            label1.Text = Nombre;
        }
}
```





```
decimal Valor = 135.20M;
label2.Text = Valor.ToString("#,##0.00");

DiasSemana Dia = DiasSemana.Martes;
label3.Text = Dia.ToString();
}
}
}
```

En este ejercicio examine cada línea de código y luego responda la pregunta. Asuma que cada código se ejecuta en secuencia como está escrito.

```
1.
Código:
int x = 10:
int y = x++;
Pregunta:
¿Cuál es el valor de y? ¿Por qué?
Código (continua del anterior)
x += 10;
Pregunta
¿Cuál es el valor de x? ¿Por qué?
Código
int z = 30;
int a = x + y * z;
Pregunta
¿Cuál es el valor de a? ¿Por qué? Escribir esto de una forma más legible.
Código (continua del anterior)
int a = 10:
int b = a++:
bool myBool = (a == b);
Pregunta
```

#### Pasos para verificar las respuestas con una aplicación.

¿Qué hace este código? ¿Cuál es el valor de myBool?

- 1. Abrir una nueva instancia de Visual Studio 2008.
- 2. En el menú **File**, hacer click en **New**, y luego en **Project**. o hacer click en **Create Project** de la ventana **Start Page**.
- 3. Aparecerá el cuadro de diálogo New Project.
- Para elegir el tipo de proyecto, expandir el nodo Visual C# y hacer click en Windows, seleccionar el template Windows Forms Application.
- 5. Especificar un nombre y una ubicación para el proyecto y presionar **OK**. Visual Studio creará el proyecto.
- 6. Para agregar el botón, desde la ventana **Toolbox** arrastar un control **Button** al formulario. Presionar doble click sobre el button para crear el código del evento Click.



SISTEMAS Av de Mayo 676 Cap Fed



- En dicho evento copiar el código correspondiente y verificar las respuestas realizando breakpoints con las herramientas de debug y siguiendo el código paso a paso.
- 8. También puede modificar el código y agregar un **MessageBox** de manera de ir mostrando los resultado.
- 9. Probar el ejercicio

```
private void button1_Click(object sender, EventArgs e)
            // 1
            int x = 10;
            int y = x++;
            MessageBox.Show("x: " + x.ToString());
            MessageBox.Show("y: " + y.ToString());
            // 2
            x += 10;
            MessageBox.Show("x: " + x.ToString());
            // 3
            int z = 30;
            int a = x + y * z;
            MessageBox.Show("z: " + z.ToString());
            MessageBox.Show("a: " + a.ToString());
            // 4
            a = 10;
            int b = a++i
            bool myBool = (a == b);
            MessageBox.Show("a: " + a.ToString());
            MessageBox.Show("b: " + b.ToString());
            MessageBox.Show("myBool: " + myBool.ToString());
```

#### Ejercicio 3

En este ejercicio completará un código al cual le falta una lógica condicional apropiada. Suponga que un zoológico necesita 5000 visitantes por semana para completar un proyecto. Usted usará una sentencia **if** para verificar el número de visitantes y escribir un mensaje indicando si el mismo fue por debajo o por encima de la meta de 5000 visitantes.

# **Pasos**

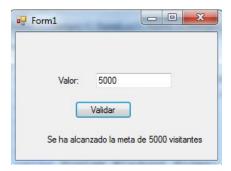
- 1. Abrir una nueva instancia de Visual Studio 2008.
- 2. En el menú File, hacer click en New, y luego en Project. o hacer click en Create Project de la ventana Start Page.
- 3. Aparecerá el cuadro de diálogo New Project.
- Para elegir el tipo de proyecto, expandir el nodo Visual C# y hacer click en Windows, seleccionar la plantilla Windows Forms Application.
- Especificar un nombre y una ubicación para el proyecto y presionar OK. Visual Studio creará el proyecto.
- Agregar una etiqueta al formulario, seleccionando un componente Label. La etiqueta debe tener una leyenda que dice "Valor"



SISTEMAS Av de Mayo 676 Cap Fed



- Agregar un cuadro de texto al formulario, seleccionando en este caso un componente TextBox. Ponerlo al lado de la etiqueta ingresada en el punto anterior.
- 8. Agregar un botón al formulario, arrastrar un componente **Button** al formulario. Modificar la propiedad **Text** del botón a "Validar".
- 9. Agregar una segunda etiqueta cuya leyenda este vacía.
- 10. Acomodar los controles de manera que el formulario quede parecido a la imagen que aparece a continuación
- 11. Para dar funcionalidad al botón, hacer doble click sobre el mismo. Agregar el código necesario para que el método **button1\_click** valide si el valor ingresado en el cuadro de texto es mayor o igual a 5000 y muestre el resultado en la etiqueta:"Se ha alcanzado la meta de 5000 visitantes", o de lo contrario, "No se ha alcanzado a la meta".
- 12. Probar el ejercicio ingresando valores menos a 5000, 5000 exacto o mayores a 5000. Verifique que el mensaje sea el correcto



```
private void button1_Click(object sender, EventArgs e)
{
    if (System.Convert.ToInt32(textBox1.Text) >= 5000)
        label2.Text = "Se ha alcanzado la meta de 5000 visitantes";
    else
        label2.Text = "No se ha alcanzado a la meta";
}
```

# Ejercicio 4

En este ejercicio, se usará una iteración **for** para calcular la suma de los enteros del 1 al 1000. Luego deberá armar el mismo ejemplo usando una sentencia **while** 

#### Pasos

- 1. Abrir una nueva instancia de Visual Studio 2008.
- 2. En el menú File, hacer click en New, y luego en Project. o hacer click en Create Project de la ventana Start Page.
- 3. Aparecerá el cuadro de diálogo New Project.
- 4. Para elegir el tipo de proyecto, expandir el nodo **Visual C#** y hacer click en **Windows**, seleccionar la plantilla **Windows Forms Application**.
- 5. Especificar un nombre y una ubicación para el proyecto y presionar **OK**. Visual Studio creará el proyecto.



SISTEMAS Av de Mayo 676 Cap Fed



- Agregar un botón al formulario. Modificar la propiedad Text del botón a "Calcular" en la ventana Properties.
- 7. Agregar una etiqueta al formulario. La etiqueta se usará para mostrar el resultado.
- 8. Para dar funcionalidad al botón, hacer doble click sobre el mismo. Agregar el código necesario para que el método **button1\_click** sume los enteros del 1 al 1000 con una sentencia **for** y muestre el resultado obtenido en la etiqueta.
- 9. Probar el ejercicio
- 10. Presionar el botón "Calcular" y verificar que el mensaje en la etiqueta sea el correcto.
- 11. Agregar otro botón y otra etiqueta al formulario y realizar la misma funcionalidad que en el punto 9 usando una sentencia **while** en lugar de una sentencia **for**.

```
private void button1_Click_1(object sender, EventArgs e)
{
   int Sumar = 0;
   for (int i = 1; i <= 1000; i++)
   {
      Sumar += i;
   }
   label1.Text = Sumar.ToString("#,##0.00");
}

private void button2_Click_1(object sender, EventArgs e)
{
   int Sumar2 = 0;
   int valor = 1;
   while (valor <= 1000)
   {
      Sumar2 += valor;
      valor++;
   }
   label2.Text = Sumar2.ToString("#,##0.00");
}</pre>
```

# Ejercicio 5

En este ejercicio se creará una aplicación para convertir datos expresados en grados Fahrenheit a Celsius y viceversa.

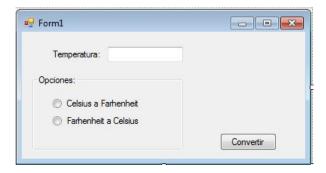
#### **Pasos**

- 1. Abrir una nueva instancia de Visual Studio 2008.
- 2. En el menú File, hacer click en New, y luego en Project. o hacer click en Create Project de la ventana Start Page.
- 3. Aparecerá el cuadro de diálogo New Project.
- 4. Para elegir el tipo de proyecto, expandir el nodo Visual C# y hacer click en Windows, seleccionar la plantilla Windows Forms Application.
- 5. Especificar un nombre y una ubicación para el proyecto y presionar **OK**. Visual Studio creará el proyecto.

1 1

6. Diseñar el siguiente formulario:





7. En el evento **Click** del botón Convertir generar la conversión, solo cuando el textbox tiene ingresado algún valor, teniendo en cuenta los siguientes cálculos:

```
Celsius = 5 * (Temperatura - 32) / 9
Fahrenheit = 9 * Temperatura / 5 + 32
```

- 8. Mostrar el resultado usando un MessageBox
- 9. En el caso de que la temperatura no se haya ingresado, mostrar un mensaje indicando que ingrese el valor
- 10. Incluir atajado de **excepciones** y probar ingresando datos erróneos.
- 11. Probar el ejercicio

#### Solución:

```
private void btnConvertir_Click(object sender, EventArgs e)
    try
        if (txtTemperatura.Text != "")
            decimal dResultado;
            decimal dTemperatura =
System.Convert.ToDecimal(txtTemperatura.Text);
            if (rdbFarCel.Checked)
                dResultado = 5 * (dTemperatura - 32) / 9;
            else
            {
                dResultado = 9 * dTemperatura / 5 + 32;
        MessageBox.Show(dResultado.ToString("#,##0.00"), "Resultado",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        Else
            MessageBox.Show("Debe ingresar el valor de la
temperatura");
    catch (Exception ex)
        MessageBox.Show(ex.ToString());
```

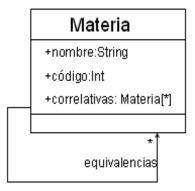


676 Cap Fed Tel // (5411) 5273-4400

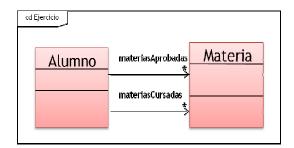
# Módulo III: Diagramas de Clases

# Ejercicio 1

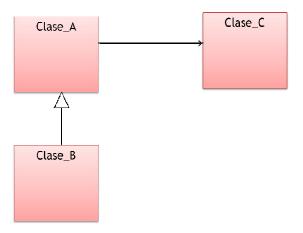
a) Indicar si el siguiente diagrama es correcto



b) Indicar si el siguiente diagrama es correcto



c) Una instancia de B conoce instancias de C







- d) Queremos modelar una cuenta de un banco: La cuenta pertenece a un único titular y de ella podemos conocer cuál es su saldo, además de poder extraer y depositar montos de dinero
- e) Modifique el ejemplo anterior de manera de poder manejar los datos del titular como Nombre y Apellido, DNI y Domicilio
- f) Modifique el ejemplo anterior de manera que una cuenta pueda tener más de un titular
- g) Ahora tenemos, para las cuentas que estamos modelando, cajas de ahorro y cuentas corrientes, que tienen el mismo comportamiento que las cuentas que teníamos en el ejercicio anterior, con la excepción de que las cajas de ahorro nunca pueden tener su saldo menor a 0.

- a) Correcto
- b) Correcto
- c) Si, una clase hija puede ocupar el lugar del padre. B es derivada de A
- d)

| CuentaBancaria |   |  |
|----------------|---|--|
| saldo          | uenta: integer<br>o: float<br>r: string                 |  |
| +dep           | aer (monto: float)<br>ositar (monto: float<br>lo: float |  |

e)

| CuentaBancaria   | Titular  |
|--|--|
| nroCuenta: integer<br>saldo: float<br>titular: string                | dni: string<br>nyAp: string<br>domicilio: string |
| +extraer (monto: float)<br>+depositar (monto: float<br>+saldo: float |  |

f)

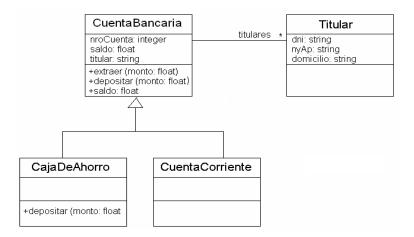
| CuentaBancaria   | tituloroo   | Titular  |
|--|-------------|--|
| nroCuenta: integer<br>saldo: float<br>titular: string                | titulares * | dni: string<br>nyAp: string<br>domicilio: string |
| +extraer (monto: float)<br>+depositar (monto: float<br>+saldo: float |             |  |

g)



76 Cap Fed Tel // (5411) 5273-4400

1 1

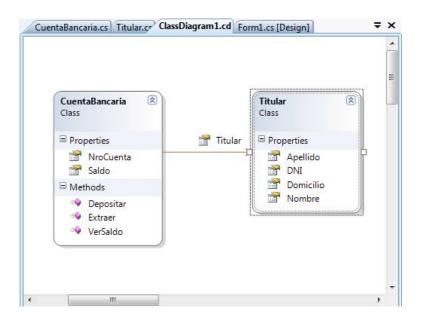


Armar el diagrama del punto e (del ejercicio anterior), usando el **View Class Diagram** dentro del Visual Studio

#### **Pasos**

- 1. Abrir una nueva instancia de Visual Studio 2008.
- 2. En el menú **File**, hacer click en **New**, y luego en **Project**. o hacer click en **Create Project** de la ventana **Start Page**.
- 3. Aparecerá el cuadro de diálogo New Project.
- 4. Para elegir el tipo de proyecto, expandir el nodo **Visual C#** y hacer click en **Windows**, seleccionar la plantilla **Windows Forms Application**.
- 5. Presionar el icono del **View Class Diagram** que se encuentra en la parte superior derecha del **Solution Explorer**
- 6. Borrar las clases que aparecen automáticamente al crear el diagrama
- Crear una nueva clase haciendo botón derecho sobre el diagrama Add Class. Llamarla CuentaBancaria
- 8. Agregue a la misma 3 propiedades: NroCuenta (tipo de dato int), Saldo (tipo de dato decimal) y Titular (tipo de dato string)
- Agregue también 3 métodos: Extraer (tipo void), Depositar (tipo void), VerSaldo (tipo decimal).
- Abra la clase CuentaBancaria.cs que se creó desde el diagrama. Analice el código escrito en la misma
- Cree una nueva clase dentro del diagrama de la misma forma que el punto 7. El nombre del clase debería ser Titular
- 12. Agregue 4 propiedades: DNI, Nombre, Apellido y Domicilio, todas de tipo string
- 13. En la clase CuentaBancaria cambiar el tipo de la propiedad Titular a Titular (esta es la clase creada en el punto 11, estamos haciendo que una propiedad sea del tipo de esta clase)
- 14. Sobre la propiedad Titular de la clase CuantaBancaria hacer botón derecho y seleccionar Show as Association. Esto debería mostrar la línea que representa la asociación







# Módulo IV: POO con C#

#### Ejercicio 1

En este ejercicio trabajará con clases, propiedades y Métodos.

#### Pasos:

- 1. Abrir Visual Studio 2008.
- 2. En el menú File, hacer click en New, y luego en Project.
- 3. Aparecerá el cuadro de diálogo New Project.
- 4. Para elegir el tipo de proyecto, expandir Visual C# y hacer click en Windows, seleccionar la plantilla Aplicación Windows Forms.
- Especificar un nombre y una ubicación para el proyecto y presionar OK. Visual Studio creará el proyecto.
- Agregar una biblioteca de clases a la solución. En el menú File, hacer click en Add y luego en New Project. Expandir Visual C# y hacer click en Windows, seleccionar la plantilla Class Library.
- 7. En la ventana Solution Explorer, hacer click con el botón secundario del mouse sobre el nodo References perteneciente al proyecto del formulario y seleccionar Add Reference. En el cuadro de diálogo, hacer click sobre la solapa Projects y seleccionar el proyecto de tipo Biblioteca de Clases, presionar OK.
- Dentro del proyecto Biblioteca de Clase, elimine la clase Class1 que se creó automáticamente y agregue una nueva clase llamada Autenticacion. Para crearla, sobre el nombre del proyecto en el Explorador de Soluciones botón derecho – Add – Class.
- 9. Agregar en esa clase 2 propiedades llamadas Usuario y Contraseña de tipo string. En el caso del Usuario validar que la misma no reciba un valor vacio y en el caso de la contraseña validar que el mínimo del dato ingresado sean 5 caracteres. En ambos casos disparar un error en el caso de no cumplirse con las especificaciones.
- 10. Definir la siguiente función: Nombre: **Autenticar**, Sin Parámetros y que devuelva un tipo de dato verdadero / falso
- 11. Dentro del mismo escribir un código que compare los valores de las propiedades Usuario y Contraseña con valores declarados en el mismo método. O sea, declare dos variables internas para usuario y contraseña válidos, asígneles algún valor y compárelos con las propiedades. Si ambos coinciden devolver el valor "true" sino devolver "false".
- 12. Sobre el formulario Form1 de la aplicación Windows que fue creado automáticamente por la aplicación, agregar los controles de manera que el mismo quede como la imagen que se muestra a continuación:



13. Generar el evento Click para el botón Aceptar. Dentro de este botón generar un código que instancie la clase Autenticación, llene las propiedades con los valores de los textbox e invoque al método Autenticar.



676 Cap Fed Tel // (5411) 5273-4400



- 14. Mostrar en un MessageBox si la autenticación fue exitosa o no
- 15. Probar el ejercicio

#### Clase

```
public class Autenticacion
      private string _Usuario;
      private string _Contraseña;
      public string Usuario
        {
            get { return _Usuario; }
            set {
                  if (value == "")
                    throw new Exception("Debe ingresar el usuario")
                     _Usuario = value;
            }
        }
        public string Contraseña
            get{ return _Contraseña; }
            set{
                if (value.Length < 5)</pre>
                    throw new Exception("El mínimo de caracteres para
la contraseña es 5");
                else
                    _Contraseña = value;
            }
        public bool Autenticar()
            string sUsuario = "Usuario1";
            string sContraseña = "12345";
            if (Usuario == sUsuario && Contraseña == sContraseña)
                return true;
            else
                return false;
```

#### **Formulario**

```
private void btnAceptar_Click(object sender, EventArgs e)
{
    Try
    {
```



676 Cap Fed Tel // (5411) 5273-4400



```
ClaseAutenticacion.Autenticacion obj = new
ClaseAutenticacion.Autenticacion();
    obj.Usuario = txtUsuario.Text;
    obj.Contraseña = txtContras.Text;
    if (obj.Autenticar())
        MessageBox.Show("Datos Correctos");
    Else
        MessageBox.Show("El usuario y/o la contraseña son
incorrectos");
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

En este ejercicio modificará el ejercicio anterior para probar otros temas

#### **Pasos**

- 1. Abrir el ejercicio anterior.
- 2. En la clase **Autenticación** crear una **sobrecarga** del método **Autenticar** de manera que reciba 2 parámetros: usuario y contraseña ambos strings
- 3. En el código de esta sobrecarga validar usando los parámetros en vez de las propiedades.
- 4. Modificar el código del botón Aceptar para que invocar a esta nueva sobrecarga
- 5. Probar el ejercicio

# Solución

# Clase

```
public bool Autenticar(string pUsuario, string pContraseña)
{
   string sUsuario = "Usuariol";
   string sContraseña = "12345";
   if (pUsuario == sUsuario && pContraseña == sContraseña)
      return true;
   else
      return false;
}
```

### **Formulario**



SISTEMAS Av de Mayo 676 Cap Fed



```
MessageBox.Show("El usuario y/o la contraseña son
incorrectos");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

En este ejercicio modificará el ejercicio anterior para probar otros temas

#### **Pasos**

- 1. Abrir el ejercicio anterior.
- En la clase Autenticación crear un constructor que reciba dos parámetros: Usuario y Contraseña. En el código inicializar las propiedades usando los parámetros recibidos.
- 3. Modificar el código del botón Aceptar para usar el constructor de manera de inicializar las propiedades con los valores de los textbox e invoque a la sobrecarga que se programó en el ejercicio 1
- 4. Probar el ejercicio

#### Solución

#### Clase

```
public Autenticacion(string pUsuario, string pContras)
{
    Usuario = pUsuario;
    Contraseña = pContras;
}
```

# **Formulario**

```
private void btnAceptar_Click(object sender, EventArgs e) {
    try
    {
        ClaseAutenticacion.Autenticacion obj = new
ClaseAutenticacion.Autenticacion(txtUsuario.Text, txtContras.Text);
        if (obj.Autenticar())
            MessageBox.Show("Datos Correctos");
        else
            MessageBox.Show("El usuario y/o la contraseña son
incorrectos");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```





En este ejercicio trabajará con herencia

#### Pasos:

- 1. Abrir Visual Studio 2008.
- 2. En el menú File, hacer click en New, y luego en Project.
- 3. Aparecerá el cuadro de diálogo New Project.
- 4. Para elegir el tipo de proyecto, expandir **Visual C#** y hacer click en **Windows**, seleccionar la plantilla **Aplicación Windows Forms**.
- 5. Especificar un nombre y una ubicación para el proyecto y presionar **OK**. Visual Studio creará el proyecto.
- Agregar un proyecto de tipo Windows Form Control Library a la solución. En el menú File, hacer click en Add y luego en New Project. Expandir Visual C# y hacer click en Windows, seleccionar la plantilla Windows Form Control Library. Llamar al proyecto Controles
- 7. Borrar el control UserControl1.cs creado automáticamente.
- 8. Agregar una nueva clase llamada miTextBox.
- Modificar la declaración de la clase miTextBox de manera que sea pública y que herede a la clase System.Windows.Forms.TextBox.
- 10. Esta clase contiene toda la funcionalidad de un TextBox de .Net. En este ejemplo le agregaremos funcionalidad para que pueda controlar el tipo de dato que está permitido ingresar en el control y el manejo de selección del texto al tomar el foco.
- Agregar un enum llamado eContenido (fuera de la declaración de la clase) que contenga 2 valores: "Caracteres", "Numeros".
- 12. Agregar en la clase una nueva propiedad llamada **Contenido** de tipo eContenido. Esta propiedad se usará para validar el tipo de dato aceptado en el textbox.
- 13. Agregar una segunda propiedad de tipo Bool llamada TextoSeleccionado. Esta propiedad controlará si al tomar el foco en el control el texto debe estar seleccionado o no.
- 14. Definir la siguiente función: Nombre: miTextBox\_KeyPress, Parámetros: sender , Tipo de Dato: object y e Tipo de Dato: System.Windows.Forms.KeyPressEventArgs
- 15. La firma de este método es la misma que la del atajador del evento KeyPress de un textbox ya que representa esa funcionalidad.
- 16. El código de esta función valida que si la propiedad Contenido es "Numeros" solo se reciban las teclas del 0 al 9 (ASCII del 48 al 57), el ASCII 8 que representa al BackSpace y el ASCII 45 que representa al signo menos (-). Si el contenido es carácter se aceptan todos los valores
- 17. Crear un constructor sin parámetros dentro de la clase. Dentro del constructor declarar que el método escrito en los puntos anteriores representa al evento **KeyPress** de este control (el mismo lo hereda de la clase TextBox).
- 18. Definir la siguiente función: Nombre: miTextBox\_GotFocus. Parámetros: sender, Tipo de Dato: object y e Tipo de Dato: System. EventArgs
- 19. La firma de este método es la misma que la del atajador del evento **GotFocus** de un textbox ya que representa esa funcionalidad
- 20. El código de esta función verificará el valor de la propiedad TextoSeleccionado y en el caso de ser "true" seleccionará el texto ya ingresado en el textbox.
- 21. Agregar al constructor ya definido la declaración del evento GotFocus de manera que al dispararse el mismo invoque a la función escrita en el punto anterior.
- 22. En el constructor inicializar la propiedad TextoSeleccionado en True de manera que este sea el valor por default de esta propiedad.
- 23. Compilar el control



676 Cap Fed Tel // (5411) 5273-4400

```
public enum eContenido
    Caracteres,
   Numeros
public class miTextBox: System.Windows.Forms.TextBox
    private eContenido _Contenido;
    private bool _TextoSeleccionado;
    public eContenido Contenido
       get {return _Contenido;}
       set {_Contenido = value;}
    public bool TextoSeleccionado
       get {return _TextoSeleccionado;}
       set {_TextoSeleccionado = value;}
    private void miTextBox_KeyPress(object sender,
System.Windows.Forms.KeyPressEventArgs e)
       if (this.Contenido == eContenido.Numeros)
              if (e.KeyChar >= 48 && e.KeyChar <= 57)</pre>
                    e.Handled = false;
                else if (e.KeyChar == 8)
                    e.Handled = false;
                else if (e.KeyChar==45)
                    e.Handled = false;
                else
                    e.Handled = true;
            }
            Else
                e.Handled = false;
        }
    public miTextBox()
            this.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(
miTextBox_KeyPress);
            TextoSeleccionado = true;
            this.GotFocus += new
System.EventHandler(miTextBox_GotFocus);
    }
    private void miTextBox_GotFocus(object sender, System.EventArgs e)
```



yo 676 Cap Fed Tel.// [5411) 5273-4400



```
{
    if (TextoSeleccionado)
        base.Select(0, base.Text.Length);
}
```

En este ejercicio usará el control creado en el ejercicio anterior

#### Pasos:

- 1. Abrir el ejercicio anterior.
- Una vez compilado el control creado en el punto anterior, miTextBox debería aparecer en la toolbox como cualquier otro control de .Net. Normalmente aparece en la parte superior de la toolbox.
- 3. Arrastrar 2 controles miTextbox sobre el Form1.
- 4. En la ventana de propiedades aparece la propiedad Contenido (aunque esta propiedad fue creada por nosotros aparece en la lista de propiedades exactamente igual que las propiedades de .Net). En uno de los textbox seleccionar "Caracteres" y en el otro "Numeros".
- También aparece la propiedad **TextoSeleccionado**. Verifique que la misma esté en True. Si quiere puede marcar esta propiedad en False en alguno de los dos textbox, de manera de verificar los distintos comportamientos.
- 6. Ejecutar y probar. En uno de ellos se podrá ingresar cualquier carácter y en el otro solo números.
- 7. Verifique que al pasar el foco entre ambos textbox el texto ya ingresado queda seleccionado automáticamente, si se dejo la propiedad TextoSeleccionado en True.
- 8. Probar el ejercicio





# Módulo V: XHTML y Scripts

#### Ejercicio 1

En este ejercicio trabajará con XHTML.

- 1. Abrir Visual Studio 2008.
- En el menú File, hacer click en New, y luego en Web Site. Seleccione ASP.NET Web Site. En Location deje la opción File System e ingrese un nombre para la aplicación.
- 3. Oprima el botón OK. Visual Studio creará el ambiente de trabajo para el sitio.
- 4. Aparece la página Default.aspx en la vista de diseño. Seleccionar la solapa Source
- 5. Modifique el valor del elemento title de manera que diga "Mi primera página"
- 6. En el **body**, agregue un texto: "Curso de XHTML" usando alguno de los elementos para encabezados y luego otro texto "Esta es mi primera página escrita con XHTML" usando alguno de los encabezados más chicos.
- 7. Ingrese a continuación un **hipervínculo** a la página de Microsoft. Para armar el vínculo escriba el siguiente texto: Si quiere abrir la página de Microsoft presione aquí. Solo la palabra aquí debe ser el vínculo.
- 8. Busque alguna **imagen** en la máquina e incluya el archivo en el proyecto web. Luego muéstrela en la página
- 9. Luego ingrese una lista ordenada que muestre al menos tres países
- 10. Probar el ejercicio

#### Solución

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
   <title>Mi primera página</title>
</head>
<body>
  <h1>Curso de XHTML</h1>
  <h3>Esta es mi primera página escrita en XHTML</h3>
  Si quiere abrir la página de Microsoft presione <a
href="http://www.Microsoft.com">aquí</a>
  <img src="microsoft.png" alt="imagen de MS" />
  Argentina
       Brasil
       Uruguay
   </body>
```

#### Ejercicio 2

En este ejercicio trabajará con tablas

- 1. Abrir el ejercicio anterior.
- Agregar una nueva página. Para ello hacer doble click sobre el nombre del proyecto en el Explorador de Soluciones, seleccionar Add New Item y seleccionar un elemento Web Form. Puede ponerle el nombre que desee
- 3. Inserte una tabla de 4 filas y 2 columnas, sin bordes.
- 4. En la primera fila poner los títulos Paises y Provincias respectivamente
- 5. En la segunda fila, en la celda de la izquierda escribir Argentina y en la de la derecha



SISTEMAS Av de Mayo 676 Cap Fed



Buenos Aires. En la siguiente fila, no repetir el nombre del país y en la de la derecha ingresar Córdoba. Dado que el nombre del país no lo repetimos juntar las 2 celdas de manera que Argentina quede asociada a las 2 provincias

Repetir el paso anterior para Uruguay como país y Montevideo y Saldo como provincias

#### Solución

```
<body>
  <form id="form1" runat="server">
  <div>
    <thead>
        Paises
          Provincias
        </thead>
      Argentina
         Buenos Aires
        >
          Córdoba
        Uruguay
          Montevideo
        Salto
        </div>
  </form>
</body>
```

#### Ejercicio 3

En este ejercicio trabajará con estilos.

- En el menú File, hacer click en New, y luego en Web Site. Seleccione ASP.NET Web Site. En Location deje la opción File System e ingrese un nombre para la aplicación.
- 2. Oprima el botón OK. Visual Studio creará el ambiente de trabajo para el sitio.
- 3. Si no quiere trabajar con un nuevo proyecto puede crear una nueva página en el proyecto de los ejercicios anteriores
- 4. En este ejercicio se trabajará con una hoja de estilos. Para ello hacer doble click sobre el nombre del proyecto en el Explorador de Soluciones, seleccionar Add New Item y seleccionar un Style Sheet. Puede ponerle el nombre que desee
- 5. En el head del html de la página Default.aspx agregue el **link** necesario para relacionar la hoja de estilo con esta página



SISTEMAS Av de Mayo 676 Cap Fed



- 6. Arme los siguientes estilos:
- 7. Para el elemento la letra sea de color rojo
- 8. Para el elemento <h1> que el alineado este centrado y sea de color azul
- 9. Para el elemento <a> que para el link sea verde, que cuando el vínculo ya fue visitado sea púrpura y cuando se pasa por encima con el mouse sea azul y subrayado.
- 10. Agregue un párrafo, un título y un vinculo en la página Default.aspx
- 11. Probar el ejercicio

#### Hoja de Estilo

```
/* Este estilo convertirá el color
del texto del párrafo en rojo*/
p {color:Red;}
/* Este estilo alineará en el centro
el texto del tag hl y lo mostrará en color azul*/
hl {text-align:center; color:Blue}
/* Estos estilos controlan la presentación
y colores de un hipervínculo.*/
a:link {text-decoration:none; color:Green}
a:visited {text-decoration:none; color:Purple;}
a:hover {text-decoration:underline;color:blue}
```

#### Página

#### Ejercicio 4

En este ejercicio trabajará con códigos javascript

#### **Pasos**

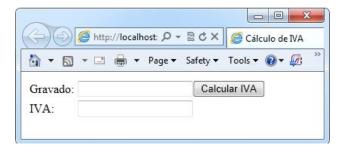
- En el menú File, hacer click en New, y luego en Web Site. Seleccione ASP.NET Web Site. En Location deje la opción File System e ingrese un nombre para la aplicación.
- 2. Oprima el botón OK. Visual Studio creará el ambiente de trabajo para el sitio.
- 3. Si no quiere trabajar con un nuevo proyecto puede crear una nueva página en el proyecto de los ejercicios anteriores
- En la página Default.aspx creada automáticamente ingrese una tabla con 2 filas y 2 columnas.



SISTEMAS Av de Mayo 676 Cap Fed



- En la primera fila ingrese el texto Gravado: y en la segunda columna una textbox y un button (input button). Renombre el textbox a txtGravado y el botón e btnCalcular. El botón debería tener la leyenda Calcula IVA
- 6. En la segunda fila ingrese el texto IVA: y en la segunda columna un textbox. Renombre el textbox a txtIVA
- 7. La página debería quedar parecida a la imagen que se muestra más adelante
- 8. Escriba una función en **javascript** llamada Calcular que calcule el valor del IVA multiplicando el gravado ingresado en el textbox y multiplicándolo por 0.21. Guarde el resultado en el textbox del IVA
- 9. Escriba el atajado del evento onclick del botón de manera que invoque a esta función
- 10. Probar el ejercicio



```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Cálculo de IVA</title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
       Gravado:
              <input id="txtGravado" type="text" /><input</pre>
id="btnCalcular" type="button" value="Calcular IVA"
onclick="Calcular()" />
          IVA:
                  <input id="txtIVA" type="text" />
          </div>
   </form>
</body>
<script type="text/javascript" >
   function Calcular() {
       var Gravado = document.getElementById("txtGravado").value ;
```



876 Cap Fed Tel // [5411] 5273-4400

```
var IVA = parseFloat(Gravado) * 0.21;
       document.getElementById("txtIVA").value = IVA;
</script>
</html>
```

