Facultad: Ingeniería
Escuela: Computación
Asignatura: Programación Orientada a Objetos

GUIA 4:
Entorno de Desarrollo Visual Studio. Parte II.

Competencia

Desarrolla sistemas de información informáticos mediante la integración de principios matemáticos, ciencia computacional y prácticas de ingeniería, considerando estándares de calidad y mejores prácticas validadas por la industria del software

Materiales y Equipo

- Guía de laboratorio # 4 Programación Orientada a Objetos.
- Visual Studio C#
- Dispositivo de almacenamiento (opcional)

Introducción Teórica

INTRODUCCIÓN A WINDOWS FORM

Esta guía tiene como objeto dar seguimiento a la construcción de aplicaciones básicas de Windows Forms usando varios de los componentes más comunes que son una característica de la mayoría de las aplicaciones GUI.

Se verá como realizar rutinas básicas pero útiles a la hora de estar desarrollando aplicaciones en entorno gráfico

Procedimiento

G4_Ejemplo_01: ***Llamada de un formulario desde otro formulario y paso de variables.***

Para la creación del proyecto, en Visual Studio

1. Inicie Visual Studio.NET
2. En el menú "Archivo (FILE)" , seleccione "Abrir (OPEN)" y después seleccione la opción "Proyecto (PROJECT)"
3. Utilice como base **G3_Ejemplo_01** que desarrolló la guía anterior.

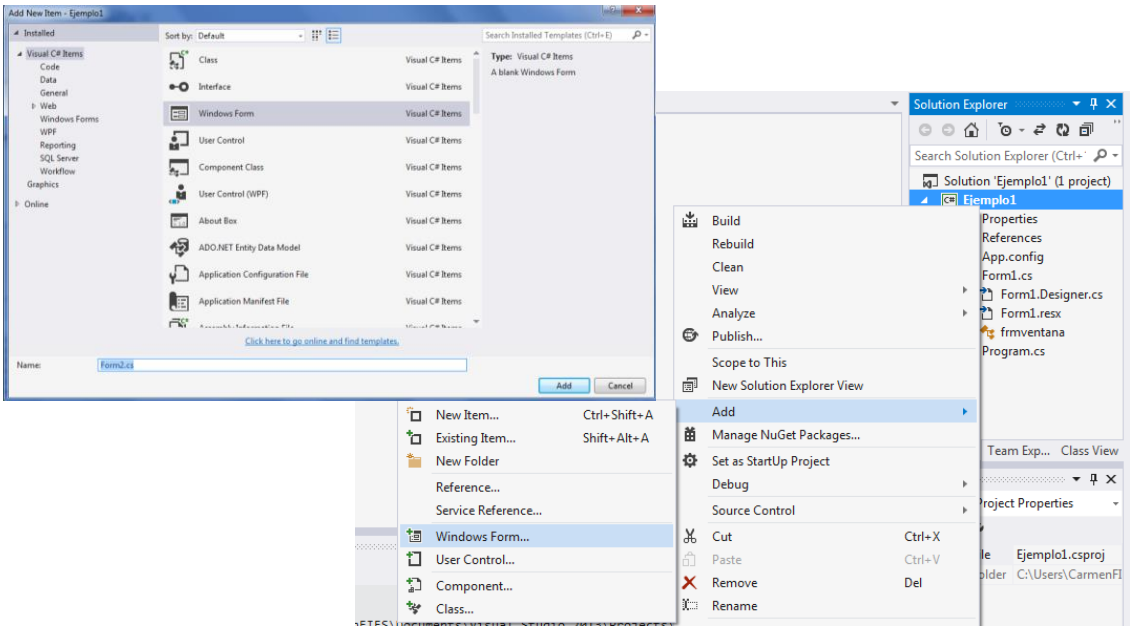
Una vez cargado el proyecto, haremos que cambie el envío de mensaje y que ya no sea por medio de un messageBox su visualización sino mediante otro formulario, para ello haremos lo siguiente:

i. Al botón Guardar Nombre le editaremos la propiedad de texto y haremos que ahora diga Enviar Mensaje.

ii. Dentro del código a la variable string nombre (que se encuentra en la modificaremos por un string texto

iii. Comentamos la línea de código del messageBox (o la eliminamos, eso es opcional, pero perderá ese código)

iv. Damos clic derecho en el nombre del proyecto y agregamos un formulario de Windows (Windows Form) (según imagen)



v. En el nuevo formulario agregaremos un label y un botón, el primero que permitirá leer el mensaje enviado y el segundo nos retornará a la primera ventana. (Quedará algo como la imagen) NO OLVIDE DARLE UN NOMBRE PERSONALIZADO A LOS ELEMENTOS.

vi. Regresemos al formulario 1 (Ventana 1) y en el código del botón para enviar, codificaremos lo siguiente:

```
string texto = txtnombre.Text;
string mensaje = string.Format("Bienvenido al segundo formulario " + texto);
Form2 frmrecibe = new Form2(mensaje); /* creo un objeto del segundo formulario,
                                         adonde mando información*/
frmrecibe.Visible = true; // muestra el nuevo formulario
this.Visible = false; // esconde el formulario actual
```

vii. En el código del segundo formulario (Ventana 2, el que acabamos de agregar) vamos a incluir un constructor adicional (con parámetros) así:

```
public Form2(string textx)
        {
            InitializeComponent();
            lbrecibido.Text = textx; // Asignamos lo recibido al label

        }
```

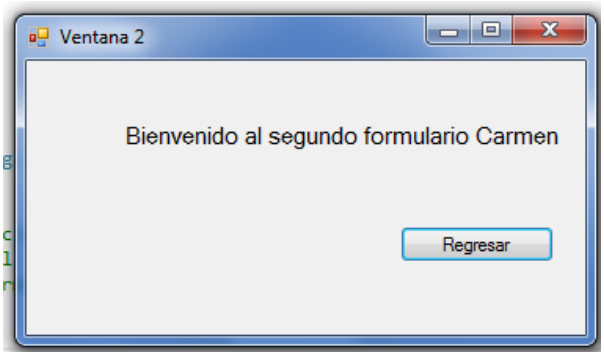
viii. En el evento click del botón que regresa a la pantalla anterior (siempre en el segundo formulario) tendríamos esto:

```
private void regreso_Click(object sender, EventArgs e)
        {

            frmventana form1 = new frmventana(); //instanciamos al primer formulario
             this.Close(); //cerramos el formulario actual
            form1.Visible = true; //hacemos visible al form1 de nuevo

        }
```

Si todo funciona bien debemos ver una pantalla como esta



G4_Ejemplo_02: *Creación de login con archivo de texto.*

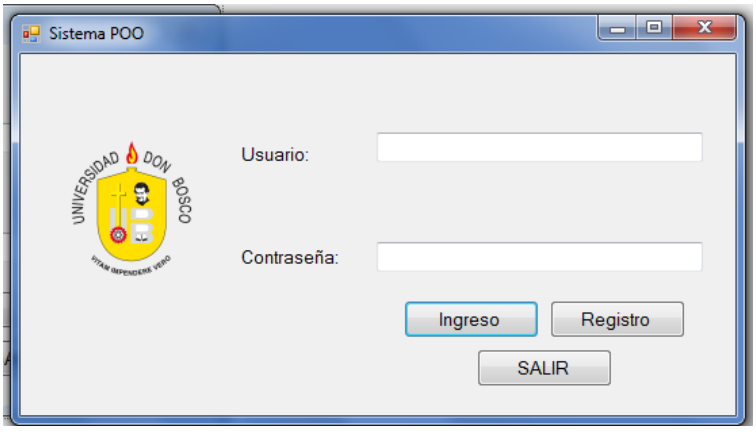
Para la creación del proyecto, en Visual Studio

1. Inicie Visual Studio.NET

2. En el menú "Archivo (FILE)" , seleccione "NUEVO (NEW)" y después seleccione la opción "Proyecto (PROJECT)"

3. Cree una carpeta con el nombre que seleccione (puede ser su número de carnet, guarde la ruta de ubicación pues será usada en el programa)

4. Cree un proyecto llamado "LoginBasico", renombre el formulario como sistema



5. Para crear este formulario se han utilizado: 3 button, 2 textboxt, 2 label, 1 pictureBox. En base a ellos modifique las propiedades que considere convenientes de forma que tenga una pantalla similar a la del ejemplo.

Para la contraseña debe modificar una propiedad: ***PasswordChar*** *e ingrese el carácter que desea, puede ser un *, esto para que no se vea la contraseña.*

Para el pictureBox si la imagen es más grande puede ajustarlo con en Size Mode: Stretch Image.

6. En el código (F7) crearemos una variable de tipo string llamada password (ésta pertenece a la clase pero no a ningún método en específico)

```
string password;
```

En el botón registro codificaremos lo siguiente:

```
private void btnregistro_Click(object sender, EventArgs e)
{
        string usuario = txtuser.Text;
        string contra = txtpass.Text;
          string url= "C:\\POO\\" + usuario +".txt"; //usted elige ubicación de carpeta, la
          que hizo en el paso 3, pero esta debe existir

        if(File.Exists(url)) //verifica que el archive exista
        {
            MessageBox.Show("ERROR. ¡Usuario ya existe!"); //usuario registrado
            txtuser.Clear(); //limpiamos todos los textbox
            txtpass.Clear();
        }
        else
        {
           File.WriteAllText(url, contra); /*Crea un Nuevo archivo con ese nombre y guarda
          dentro del archivo el valor del segundo parámetro*/
            MessageBox.Show("Usuario Registrado con éxito");
```

```
            txtuser.Clear();
            txtpass.Clear();

           }

}
```

7. Para el botón de ingreso, su código tendrá:

```
private void btningreso_Click(object sender, EventArgs e)
 {
     string usuario = txtuser.Text; //capturamos los valores de usuario y contraseña
     string contra = txtpass.Text;

     string url= "C:\\POO\\" + usuario +".txt";

     if(File.Exists(url)) //verifica si existe
     {
         password = File.ReadAllText(url); //lee el texto almacenado dentro del archivo
         if(contra.Equals(password)) //verifica si contraseña es igual al archivo
         {
            MessageBox.Show("¡Ingreso exitoso, bienvenido!"); //login exitoso
         }
         else
         {
              MessageBox.Show("¡Contraseña incorrecta! ☹ ");//login fallido
         }
     }
     else
     {
          MessageBox.Show("¡Usuario incorrecto! ☹"); //usuario incorrecto
     }

    }
```

8. Finalmente vamos a codificar el SALIR

```
      private void btnsalir_Click(object sender, EventArgs e)
             {
                 Application.Exit();
             }
```

9. Corremos el programa y anotamos lo que sucede

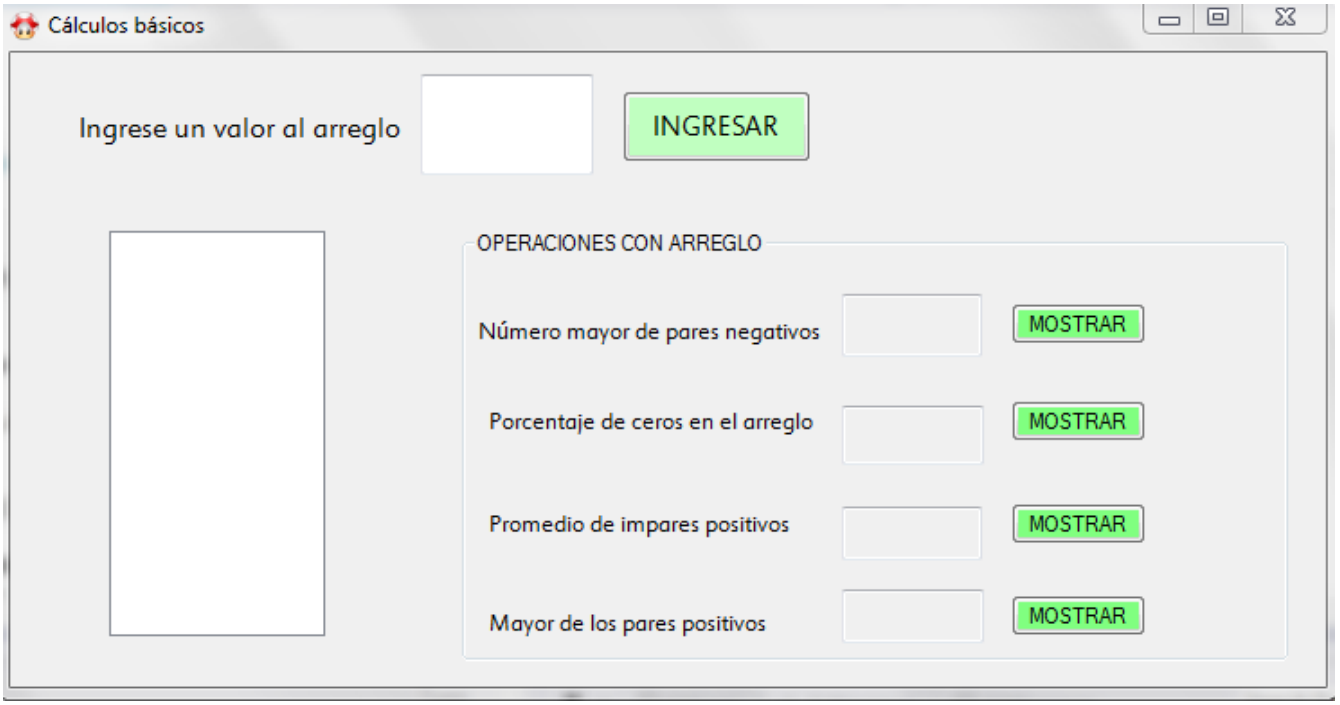
Ejercicio 2

Se pide lo siguiente:

De un listado de N enteros cualquiera brindados por usuario, determinar:

a) Mayor valor de los números pares negativos
b) porcentaje de valores ceros en el listado ingresado
c) valor promedio de los valores impares positivos
d) menor mayor de los pares positivos registrados

Para ello se creará una ventana como la siguiente



Para esta ventana se han utilizado:

5 botones - 5 textbox (4 con la propiedad ReadOnly en TRUE) – 1 ListBox – 5 label – 1 GroupBox

Recuerde activar los eventos para los objetos y proceda a programar lo siguiente:

```
//Evento para ingresar datos al arreglo
private void btnIngresar_Click(object sender, EventArgs e)
        {
            listbArreglo.Items.Add(txbNumero.Text);
            txbNumero.Clear();
            txbNumero.Focus();
        }
```

```
//Evento calcular al mayor de los pares negativos
private void btnCalc1_Click(object sender, EventArgs e)
        {
            int mayorneg = -1000;
            for (int i = 0; i < listbArreglo.Items.Count; i++)
            {
                string valor = listbArreglo.Items[i].ToString();
                int numero = int.Parse(valor);

                if (numero < 0 && numero % 2 == 0)
                {
                    if (numero > mayorneg)
                    {
                        mayorneg = numero;
                        txbCalculo1.Text = mayorneg.ToString();
                    }
                }
                else
                {
                    txbCalculo1.Text = "No hay números negativos pares";
                }
            }
        }

//Evento para calcular porcentaje de ceros
private void btnCalc2_Click(object sender, EventArgs e)
        {
            double cantidadnumeros = listbArreglo.Items.Count;

            double cantidadceros = 0;
            double porcentaje = 0;


            for (int i = 0; i < listbArreglo.Items.Count; i++)
            {
                string valor = listbArreglo.Items[i].ToString();
                int numero = int.Parse(valor);

                if (numero == 0)
                {
                    cantidadceros = cantidadceros + 1;

                }
            }
            porcentaje = (cantidadceros / cantidadnumeros) * 100;
            txbCalculo2.Text = porcentaje.ToString() + "%";
        }


//Evento para obtener el promedio de impares positivos
private void btnCalculo3_Click(object sender, EventArgs e)
        {
            double prom;
            double cantidadimpares = 0;
            double suma = 0;

            for (int i = 0; i < listbArreglo.Items.Count; i++)
            {
```

```
                string valor = listbArreglo.Items[i].ToString();
                int numero = int.Parse(valor);

                if (numero > 0 && numero % 2 != 0)
                {
                    suma = suma + numero;
                    cantidadimpares = cantidadimpares + 1;
                }
            }

            prom = suma / cantidadimpares;
            txtCalculo3.Text = prom.ToString();
        }

//Evento para mayor de pares positivos
private void btnCalculo4_Click(object sender, EventArgs e)
        {
            int mayor = 0;

            for (int i = 0; i < listbArreglo.Items.Count; i++)
            {
                string valor = listbArreglo.Items[i].ToString();
                int numero = int.Parse(valor);

                if (numero > 0 && numero % 2 == 0)
                {
                    if (numero > mayor)
                        mayor = numero;}
            }

            txbCalculo4.Text = mayor.ToString();
        }
```

Desarrollo de habilidades

1. Modifique los ejercicios de la GUÍA 3 de forma que:

   a) El sistema que almacena a todos ellos deberá ser modificado para que pida un usuario y una contraseña para su ingreso. Deberá validar que la contraseña y el usuario no queden vacíos; si están vacíos deberá mandar una alerta.

   b) Debe haber una ventana principal desde la cual pueda mandar a llamar al ejercicio de sueldos, al del convertidor de unidades y al de la cuadrática (un botón para cada uno).

   c) Personalice todos los ejercicios de forma que sean amigables con el usuario (alertas, mensajes para usuario, colores y otros).

2. Modifique el ejemplo 2 de forma que realice todos los cálculos por un único botón y que para ingresar datos al arreglo no sea necesario presionar un botón, es decir que se le pide borrar el botón ingresar y que ahora el ingreso se realice cuando el usuario presione la tecla Enter.

Ejercicios sugeridos

a) Investigue qué formas de poder hacer menús de ventanas permite C#. Tome como base sus ejercicios y modifique de forma que pueda aplicarlo en ellos.