

**INSTITUTO TECNOLÓGICO DE TLAXIACO**

**MATERIA:**

Tópicos avanzados de programación.

**DOCENTE:**

Ing. José Alfredo Román Cruz

**CARRERA:**

Ingeniería en Sistemas Computacionales

**GRUPO:**

4US

**EQUIPO:**

Sandra Gabriela Velasco Guzmán

Ángeles González Martínez

**Sprint 1: Interfaz gráfica de usuario.**

**Práctica 2: Interfaz gráfica en Visual Studio.**

**HEROICA CIUDAD DE TLAXIACO, OAXACA A 18 DE FEBRERO DE 2022.**



- I. **Práctica No. 2**
- II. **Nombre.**  
Interfaz gráfica en Visual Studio.
- III. **Competencia(s) a desarrollar.** El Estudiante **realiza** una **práctica** 2 de la usabilidad de una Interfaz gráfica en Visual Studio con sus características para su implementación.
- IV. **Lista de figuras:**
  - 1. Ilustración 1
  - 2. Ilustración 2
  - 3. Ilustración 3
  - 4. Ilustración 4
  - 5. Ilustración 5
  - 6. Ilustración 6
  - 7. Ilustración 7
  - 8. Ilustración 8

V. **Introducción.**

En este tema se aborda el tema de interfaz de usuarios, mismo que consiste en la forma de construir aplicaciones con interfaz gráfica amigable, enfocadas a problemas reales. Si un ser humano quiere interactuar con una máquina, necesita un medio para poder hacerlo. Ese medio es, precisamente, lo que se conoce como interfaz de usuario; la parte del sistema que interactúa con el usuario. La interfaz es un lugar de encuentro entre los bits y las personas. Un evento es la notificación que hace un elemento gráfico cuando el usuario interactúa con él. Por lo tanto, si se realiza alguna acción sobre algún elemento de la interfaz, se dice que ha generado un evento en dicho elemento.

La interfaz es la parte visible de las aplicaciones, siendo lo que se percibe de las mismas; por ello, cada vez se les está dando una importancia mayor y se está poniendo más cuidado en su desarrollo. La creación de interfaces de usuario es un área, dentro del desarrollo de software, que ha evolucionado mucho en los últimos años y lo sigue haciendo a día de hoy. Se podría decir que la interfaz de usuario es lo primero que se juzga de una aplicación, y si no tiene la calidad adecuada puede producir rechazo por parte del usuario.

Una interfaz puede incluso poner limitaciones en la comunicación de la máquina con el usuario. Todo aquello que no se pueda expresar a través de la interfaz se perderá, por ello, tiene gran importancia dentro del desarrollo de una aplicación el diseño de su interfaz. Un buen programa con una interfaz pobre y no adaptada a las tareas a realizar, no sirve de mucho, ya que la interfaz será más una barrera que un medio que facilite la interacción.

Hay que tener siempre presente que la interfaz de usuario determina la habilidad de la aplicación. La habilidad es una medida de uso de un producto por determinados usuarios, con el fin de alcanzar ciertos objetivos de eficiencia, efectividad y



satisfacción, todo ello dentro de un contexto de uso concreto. (Tópico avanzado programación, s.f.)

Por lo tanto, la interfaz de usuario debe ser usable, lo que implicará que el usuario consiga realizar las tareas por medio de ella de forma rápida y sencilla, y se encuentre cómodo con la aplicación. Una interfaz gráfica de usuario (GUI) es una interfaz de usuario en la que se hace uso de un entorno gráfico.

#### ❖ **Materiales indispensables en la práctica:**

- ✓ Software Visual Studio  
Link para bajar Visual Studio <https://visualstudio.microsoft.com/es/downloads/>
- ✓ Laptop.
- ✓ Internet.

#### ❖ **Procedimiento de la práctica:**

- ✓ La interfaz gráfica debe realizarse en Visual Studio para implementar controles, botones y eventos.
- ✓ La interfaz gráfica a diseñar puede ser una calculadora o algún otra Aplicación.
- ✓ La interfaz gráfica a diseñar debe reunir los requisitos de Usabilidad y diseño.
- ✓ Indicar en el reporte de práctica las características de la Usabilidad y diseño.

#### ❖ **Idea principal:**

El equipo de trabajo de tópicos avanzados de programación, en este caso optó por realizar una pequeña calculadora básica, misma aplicación cuenta con las 4 operaciones básicas (suma, resta, multiplicación y división).

A continuación se muestran los pasos seguidos para diseñar cada parte del diseño de la interfaz de la pequeña calculadora:



## ❖ Descripción paso a paso de nuestra calculadora:

1. Una vez descargado nuestro archivo ejecutable de Visual Studio, se procede con la instalación.
2. Se abre el software de Visual Studio, se procede a crear un nuevo proyecto, para esto primeramente se selecciona la plantilla de Windows Form, esto es con el fin de poder diseñar la interfaz de nuestra calculadora a nuestra manera (personalizado), aunado al lenguaje de programación c#, para poder programar la funcionalidad de nuestro pequeño software.
3. Una vez elegido estos formatos, se procede a clicar en <siguiente>, en el siguiente punto se le asigna un nombre a nuestro proyecto, y se crea el proyecto.
4. Una vez que se ha creado el proyecto, existe un apartado tanto para realizar nuestro diseño en el cual hemos creado el siguiente diseño para nuestra calculadora básica:

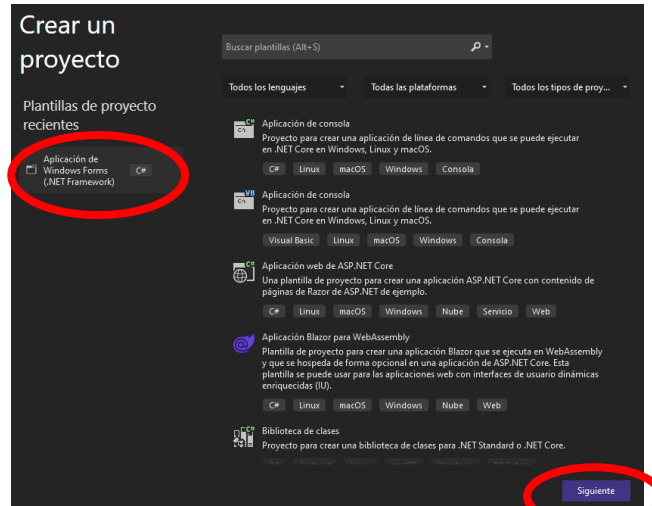


Ilustración 1. Creación del proyecto, elección de la plantilla Windows Form.

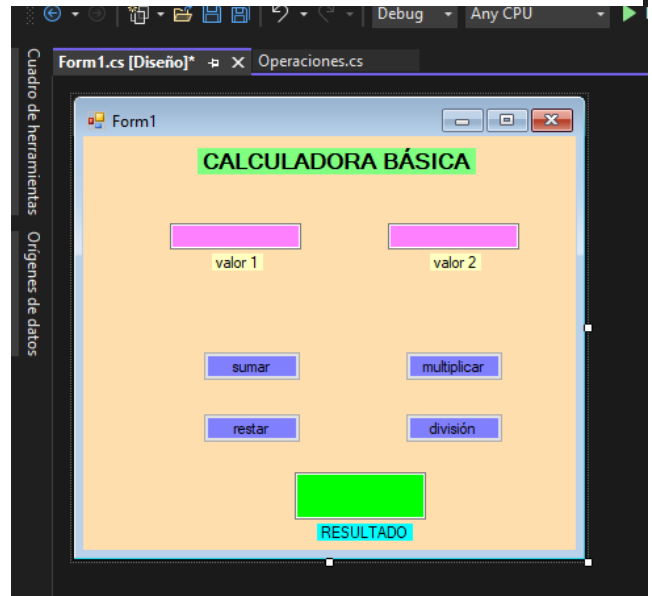
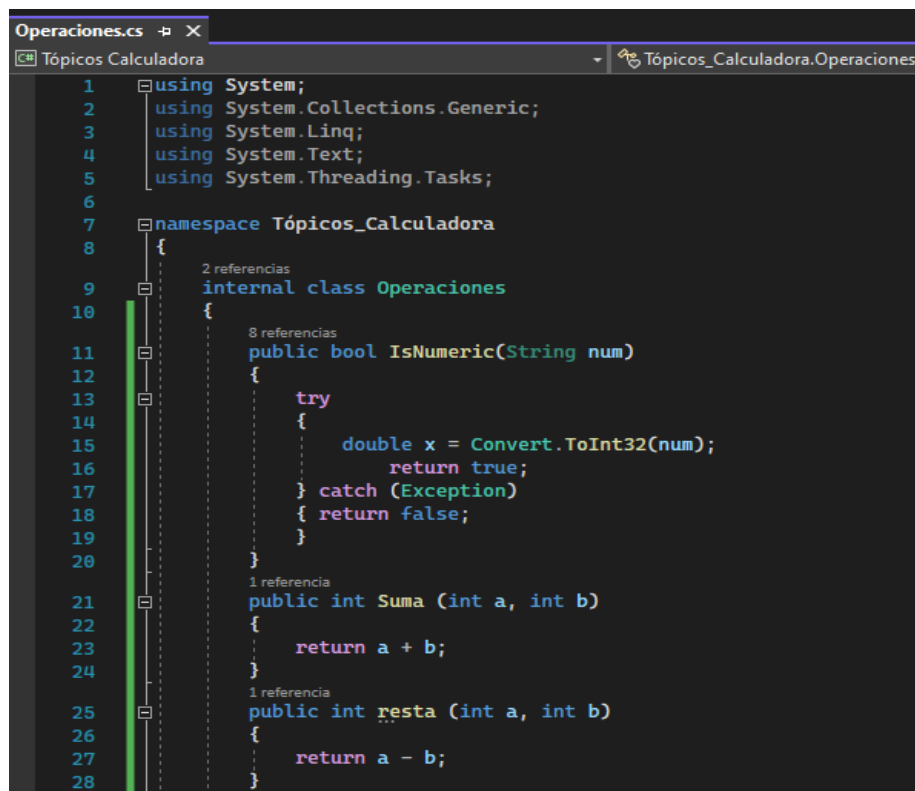


Ilustración 2. Vista de los componentes básicos que se han usado para realizar el diseño de la calculadora.

Como se observa, a nuestra calculadora le hemos diseñado los cuatro botones básicos para realizar las respectivas operaciones básicas, también se le ha insertado un text-box, el cual lleva el titulado de nuestra calculadora, lleva insertado dos “label” mismas que han sido diseñados pensando en depositar ahí los números que en su momento se quieran calcular, por último se le ha insertado un botón más, en donde la calculadora nos mostrará el resultado de nuestras operaciones.



5. Es necesario destacar que a cada uno de los elementos insertados en el diseño de la calculadora, la propiedad “name” de todos han sido renombrados, esto con el fin de que al momento de escribir nuestro código, con el cual se le dará funcionalidad a esta calculadora nos sea más factible y evitarnos la confusión.
6. Para desplegar la pestaña de nuestro código, solo se necesita darle un “doble click” a uno de nuestros botones, pero hay que ir en orden y elegir bien, porque una vez hecho esto, prácticamente la estaremos instanciando en el código.
7. Se presenta el siguiente código, esta es la que se desarrolló primeramente para por consiguiente poder implementarla en el otro código principal:

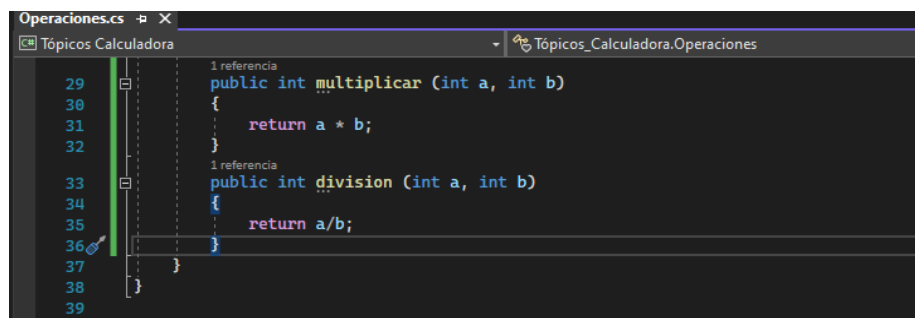


```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Tópicos_Calculadora
8  {
9      2 referencias
10     internal class Operaciones
11     {
12         8 referencias
13         public bool IsNumeric(String num)
14         {
15             try
16             {
17                 double x = Convert.ToInt32(num);
18                 return true;
19             } catch (Exception)
20             { return false; }
21         }
22         1 referencia
23         public int Suma (int a, int b)
24         {
25             return a + b;
26         }
27         1 referencia
28         public int resta (int a, int b)
29         {
30             return a - b;
31         }
32     }
33 }

```

Ilustración 3. Código Operaciones



```

29     public int multiplicar (int a, int b)
30     {
31         return a * b;
32     }
33     1 referencia
34     public int division (int a, int b)
35     {
36         return a/b;
37     }
38 }
39 }

```

- ✓ En el código que se presentó, llamado Operaciones, representa la implementación de nuestras 4 operaciones básicas, en donde también se han hecho ciertas excepciones, este código se ha desarrollado aparte, esto con la razón de que es necesario implementarlo en el código principal.

8. Se presenta el siguiente código, este es el más importante, ya que este invoca todos los métodos que se han implementado en operaciones, es de esta manera que se extrae todo de la clase Operaciones, y se implementa en esta clase principal:

```
9. using System;
10. using System.Collections.Generic;
11. using System.ComponentModel;
12. using System.Data;
13. using System.Drawing;
14. using System.Linq;
15. using System.Text;
16. using System.Threading.Tasks;
17. using System.Windows.Forms;
18.
19. namespace Tópicos_Calculadora
20. {
21.     public partial class Form1 : Form
22.     {
23.         public Form1()
24.         {
25.             InitializeComponent();
26.         }
27.         Operaciones obj = new Operaciones();
28.         private void label1_Click(object sender, EventArgs e)
29.         {
30.
31.         }
32.
33.         private void btn_sumar_Click(object sender, EventArgs e)
34.         {
35.             if (!obj.IsNumeric(txt_num1.Text) &&
36.                 !obj.IsNumeric(txt_num2.Text))
37.             {
38.                 txt_resultado.Text = "Ingrese un valor válido";
39.             }
40.             else
41.             {
42.                 int suma =
43.                     obj.Suma(Convert.ToInt32(txt_num1.Text),
44.                         Convert.ToInt32(txt_num2.Text));
45.                 txt_resultado.Text = Convert.ToString(suma);
46.             }
47.         }
48.
49.         private void btn_restar_Click(object sender, EventArgs e)
50.         {
51.             if (!obj.IsNumeric(txt_num1.Text) &&
52.                 !obj.IsNumeric(txt_num2.Text))
```



```
49.         {
50.             txt_resultado.Text = "Ingrese un valor válido";
51.         }
52.         else
53.         {
54.             int resta =
55. obj.resta(Convert.ToInt32(txt_num1.Text),
56. Convert.ToInt32(txt_num2.Text));
57.             txt_resultado.Text = Convert.ToString(resta);
58.         }
59.     }
60.     private void btn_multiplicar_Click(object sender,
61. EventArgs e)
62.     {
63.         if (!obj.IsNumeric(txt_num1.Text) &&
64. !obj.IsNumeric(txt_num2.Text))
65.         {
66.             txt_resultado.Text = "Ingrese un valor válido";
67.         }
68.         else
69.         {
70.             int mult =
71. obj.multiplicar(Convert.ToInt32(txt_num1.Text),
72. Convert.ToInt32(txt_num2.Text));
73.             txt_resultado.Text = Convert.ToString(mult);
74.         }
75.     }
76.     private void btn_division_Click(object sender, EventArgs
77. e)
78.     {
79.         if (!obj.IsNumeric(txt_num1.Text) &&
80. !obj.IsNumeric(txt_num2.Text))
81.         {
82.             txt_resultado.Text = "Ingrese un valor válido";
83.         }
84.         else
85.         {
86.             int division =
87. obj.division(Convert.ToInt32(txt_num1.Text),
88. Convert.ToInt32(txt_num2.Text));
89.             txt_resultado.Text = Convert.ToString(division);
90.         }
91.     }
92. }
```

El código que se ha presentado es aquel en el que hemos instanciado nuestra clase de operaciones, e instanciado nuestros objetos creados en la clase operaciones.





9. Una vez que se han implementado nuestros códigos, con el fin de darle funcionalidad a nuestro código, podemos ejecutar la aplicación:

Es necesario recalcar que nuestra calculadora tiene entrada de datos por teclado.

Ilustración 4. Diseño final de nuestra calculadora

## 10. Demostración de operaciones:

Ilustración 5. OPERACIÓN SUMA

Ilustración 6. OPERACIÓN MULTIPLICACIÓN

Ilustración 7. OPERACIÓN RESTA

Ilustración 8. OPERACIÓN DIVISIÓN



### **Conclusión:**

Una vez concluida esta práctica se ha observado y analizado la importancia que tienen las buenas interfaces en una aplicación, ya que esto ofrece al usuario una mejor experiencia, y también un mejor aprovechamiento de los recursos que estos contengan, las interfaces de usuario y la visualización de información es crítica para el éxito de productos y servicios que permitan a las personas tomar decisiones inteligentes rápidamente, aquí es donde se ve notoriamente la importancia que tienen las interfaces y de que sean buenos. En este caso como estudiantes se ha tratado de diseñar una pequeña aplicación con una interfaz demasiado sencilla y manejable, misma que del cual se puede deducir que el usuario no tendrá ningún problema al momento de usarla.

La principal idea de esta práctica ha sido la mediación entre hombre y máquina como se ha establecido al principio de la práctica, al realizar este tipo de prácticas se aprende demasiado y se ponen en juego las destrezas de cada estudiante, la imaginación para poder diseñar estas interfaces manejables, como equipo de trabajo se espera que la calculadora básica que se ha presentado cumpla con las expectativas esperadas.

