



Actividad | 2 | Menú en Windows

Forms

Nombre del curso

Ingeniería en Desarrollo de Software



TUTOR: Miguel Ángel Rodríguez.

ALUMNO: Uziel de Jesús López Ornelas.

FECHA: 15 de Abril Del 2025.

Tabla de Contenido

Introducción	1
Descripción	1
Justificación	2
Desarrollo	2
Diseño y Codificación de Windows Forms	2
Conclusión	14
Referencias	15

Introducción

Ya observamos en la actividad pasada el cómo utilizar de una manera sencilla Windows Forms y sus herramientas que nos permitían poder crear un formulario con diferentes características para que el usuario lograra navegar de manera cómoda y rápida, pero esta vez nos adentraremos aún más en este concepto, en la actividad anterior solo se trabajaba en una sola ventana, pero ahora serán varias las que nos tocará codificar y diseñar en donde tendrán diferentes funciones que lo harán mucho más atractivo para el usuario final. Partiremos de los conceptos previos para lograr completar con éxito los pasos que tiene la actividad y concluirla adecuadamente. Como siempre este documento tendrá las secciones correspondientes a lo que se quiere explicar en cada una de ellas, el desarrollo es en donde se tendrá todo el progreso y ejecución de la actividad en general y se agregará el archivo a GitHub para su descargue y revisión.

Descripción

Como sabemos “Windows Forms” es un software de desarrollo de aplicaciones para la plataforma de Windows que forma parte de .NET. Esta aplicación permite a los desarrolladores que lo prueban o practiquen con el crear aplicaciones de escritorio con una interfaz gráfica de usuario, por sus siglas (GUI), que, por supuesto como se ha visto en las actividades anteriores como en la que veremos a continuación está basada en botones, “TextBox”, “ComboBox”, “TabControl”, menú, etc. En donde cada uno de ellos cuenta con su propia codificación para lograr tener una funcionalidad. Es conocido por la gran facilidad de uso y su integración en otras tecnologías, como lo es el caso de C# y Visual Basic.

Un **Menú** es un elemento de la interfaz gráfica que permite a cualquier usuario acceder a diferentes funciones junto con opciones dentro de la misma aplicación, este suele aparecer en la parte superior izquierda de nuestra ventana pero que sin ningún problema con las opciones adecuadas logramos posicionarlo en las coordenadas que necesitamos. En el menú, nosotros podemos editar de manera tanto vertical como horizontalmente, este sin ningún problema al igual que cualquier elemento puede ser modificado mediante código, en la actividad de este documento se utilizará para crear una especie de índice y saltar a otras ventanas.

Justificación

Aprender a utilizar este tipo de software es fundamental para un ingeniero que está cursando la carrera en Desarrollo de Software, no solo le estas dando más herramientas para que él logre aumentar su experiencia en diferentes lenguajes de programación, sino que se adquieren aptitudes para resolver problemas, ubicar lo que está en su entorno de trabajo y saber en dónde tiene que iniciar y en donde tiene que terminar, sin embargo, la creatividad y la práctica es importante en estos campos, realizar ejercicios o jugar con el código sin duda ayuda a que el ingeniero se familiarice con el trabajo en turno. “Windows Forms”, es un programa en el que podemos jugar con eso, desde crear un simple diseño sin ninguna codificación, hasta lograr crear menús en donde el usuario logre ir de una tras otra y regresar sin ningún problema gracias a la codificación de cada formulario y los botones de acción que este tendría.

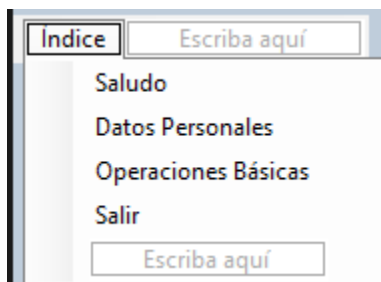
Desarrollo

Diseño y Codificación de Windows Forms

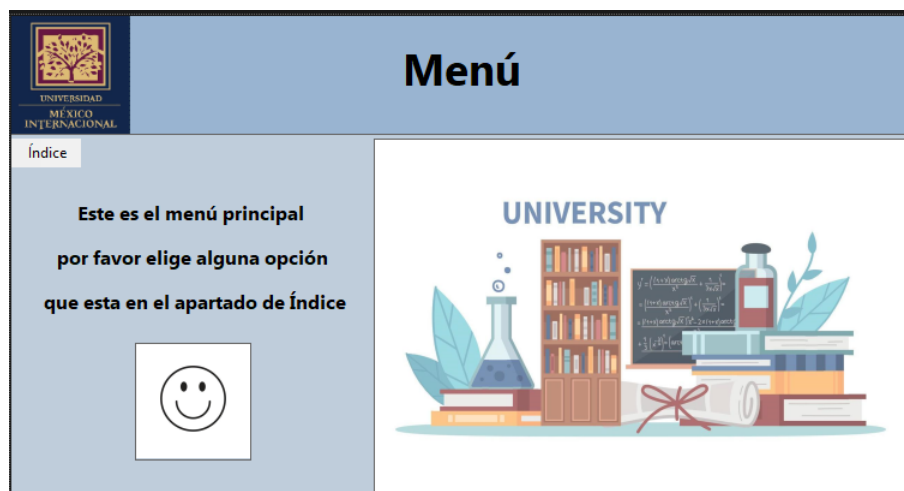
Creemos el diseño base de nuestra ventana para que se tenga una buena presentación, iniciamos con el título y la imagen, la actividad nos pide que coloquemos como título menú:



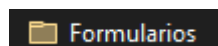
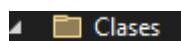
Después se crean los menús que dirán “Saludo”, “Datos Personales”, “Operaciones Básicas” y por último “Salir”:



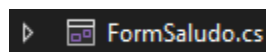
Por último, creamos el diseño restante para que se tenga una vista más atractiva para el usuario, este sería el diseño de nuestra primera ventana:



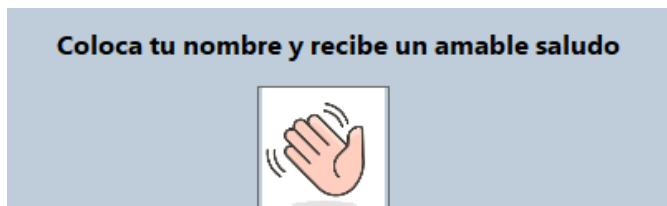
Es importante que acomodemos todo en carpetas para que no tengamos problemas a la hora de ubicar algún archivo o clase que creemos, las nombraremos como “Clases” y “Formularios”:



En la carpeta de “Formulario” es en donde crearemos los diseños de las diferentes ventanas que nos pedirá la actividad, el siguiente es el de “Saludo”, para ello seleccionamos la carpeta, damos clic en “Agregar” y después “Agregar formulario”, nos pedirá ingresar un nombre para distinguirlo y una vez hecho eso ya tenemos listo el formulario, en el el usuario tendrá que colocar un nombre para que la acción le devuelva un mensaje con su nombre, como siempre crearemos primero el diseño, el título en donde nos pide que se llame “Saludo” tal cual en nuestro menú correspondiente:



Colocamos un abrevé descripción y una imagen para que tenga una vista más agradable y no tan simple:



Colocamos una etiqueta que diga nombre y su caja de texto para que el usuario logre escribir su nombre y le devuelva el saludo tal y como nos pide la actividad:

Nombre:

Po último, se colocarán dos botones, uno de “Regresar” que nos mandara al menú principal y otro que será el de “Saludar” que mandara el mensaje correspondiente:



En este punto nos centraremos en la carpeta de “Clases”, seleccionamos dicha carpeta, damos clic derecho, “Agregar” y “Clase”, al igual que la anterior nos pedirá colocar un nombre a nuestra clase para después guardar dicho archivo, una vez dentro hacemos la codificación para que este tenga la acción que se nos pide la actividad, antes que nada, en nuestra codificación es importante colocar “public” al inicio:

```
▸ C# Clase_Saludo.cs
```

```
2 referencias
public class Clase_Saludo
{
```

Se crea un tipo de variable y una variable con la acción que se quiere dar, por ejemplo, en este caso es “Saludar” así que la colocamos, siguiente a ello entre paréntesis se coloca una variable que será la que se tomara de la caja de texto para dar el mensaje:

```
1 referencia
public string Saludar (string nombre)
```

Creamos una variable llamada “Saludo”, en ella nos mostrara un mensaje más el nombre de la persona que este colocara en la caja de texto, si esto es verdadero nos devolverá el saludo:

```
string saludo;
saludo = "Hola " + nombre + " ,¿Qué tal va tu día?";
return saludo;
```

Le daremos la acción al botón de saludar para que no haga la acción correspondiente, damos doble clic en el botón de “Saludar” para que nos mande directamente a la codificación y empezamos a trabajar en ella, el nombre que tiene el botón lo muestra el programa en la siguiente sección:

```
1 referencia
private void btn_Saludar_Click(object sender, EventArgs e)
{
```

Tenemos que mandar a llamar las variables que creamos en la clase, primero ubicar en que carpeta se encuentra, como se llama, mandar a llamar la variable, después crear una nueva instancia con el mismo nombre de la clase y nombre anterior:

```
Clases.Clase_Saludo Saludo = new Clases.Clase_Saludo();
```

Creamos una variable en la que guardara “Saludo”, la acción que colocamos en la clase que era “Saludar” en donde esta guardara la información de la caja de texto en el que el usuario colocara su nombre:

```
string Misaludo = Saludo.Saludar(txt_Nombre_Saludo.Text);
```

Por último, colocaremos una caja de texto que nos muestre dicha acción que colocamos anteriormente:

```
MessageBox.Show(Misaludo);
```

Para unir las dos ventanas tenemos que dar doble clic en la sección de “Saludo” que está en nuestro menú principal para que nos habrá una ventana en la que podemos colocar la codificación, tendrá el nombre con el que lo nombramos, por ejemplo, “Tool_Saludo_Click”:

```
1 referencia
private void Tool_Saludo_Click(object sender, EventArgs e)
{
```

Ya que estamos dentro mandamos a llamar el formulario que creamos, recuerda que es el de “Saludo” para colocar una nueva instancia con el mismo nombre y que este no la muestre en la ejecución de la consola:

```
Formularios.FormSaludo FormSaludo = new Formularios.FormSaludo();
FormSaludo.Show();
```

Ahora que si nosotros quisiéramos esconder dicha ventana solo tenemos que colocar el siguiente comando:

```
this.Hide();
```

Solo nos quedaría colocar la acción del botón “Regresar”, para ello damos doble clic en dicho botón y colocamos la codificación de que al presionarlo nos devuelva al formulario que nosotros le digamos, que nos cierre la ventana y que nos muestre el retroceso del código:

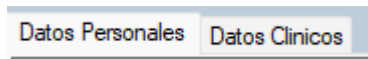
```
private void btn_Regresar_Click(object sender, EventArgs e)
{
    this.Close();
    Form_Menu back = new Form_Menu();
    back.Show();
}
```

Continuemos con el formulario de “Datos Personales”, al igual que el anterior se creará en la carpeta de Formularios. Emparemos con el diseño del título:

```
Form_Informacion_Personal.cs
```



Colocaremos dos pestañas, la primera como “Datos Personales” y la otra como “Datos Clínicos”:



En la pestaña de “Datos Personales” colocaremos el “Nombre”, “Apellido”, “Dirección”, “Ciudad” (todos los anteriores con TextBox), “Fecha de Nacimiento” (DateTimePicker), “Estado Civil” (ComboBox) y por último “Estado Civil” (RadioButton):

En la pestaña de “Datos Clínicos” se colocarán preguntas de salud con sus opciones a un lado, si es un “sí” puede proceder con la pregunta, si es un “no”, tiene que seleccionar el “ComboBox” para colocar la respuesta que crea conveniente:

The screenshot shows a web form with two tabs: 'Datos Personales' and 'Datos Clínicos'. The 'Datos Clínicos' tab is active. It contains five questions, each with 'No' and 'Si' radio buttons and a dropdown menu:

- ¿Padece de alguna enfermedad?: ☐ No ☐ Si [Dropdown]
- ¿Tienes alergias?: ☐ No ☐ Si [Dropdown]
- ¿Alguna vez te han operado?: ☐ No ☐ Si [Dropdown]
- ¿Haces ejercicio?: ☐ No ☐ Si [Dropdown]
- ¿Padece de depresión?: ☐ No ☐ Si [Dropdown]

Por supuesto tenemos el botón de “Guardar” y de “Regresar”:

The screenshot shows two buttons side-by-side. The left button is yellow with the text 'Regresar' in black. The right button is yellow with the text 'Guardar' in black.

Al igual que la anterior creamos una clase llamada como el formulario para evitar errores o confusiones:

C# Clase_Información_Personal.cs

En la clase creamos la acción que será igual a la de “Saludar” solo que aquí pondremos “Guardar” para darle sentido a la acción:

```
public class Clase_Información_Personal
{
    1 referencia
    public string Guardar (string nombre)
    {
        string guardar;
        guardar = "Gracias " + nombre + " ,los datos fueron guardados";
        return guardar;
    }
}
```

Damos doble clic en el botón de “Guardar” para colocar la acción que es idéntica a la de “Saludo”:

```
1 referencia
private void btn_Guardar_Click(object sender, EventArgs e)
{
    Clases.Clase_Información_Personal guardar = new Clases.Clase_Información_Personal();
    string Miguardado = guardar.Guardar(txt_Nombre_Info_Perso.Text);
    MessageBox.Show(Miguardado);
}
```


Unimos la ventana de “Información Personal” con la principal dando doble clic en el índice que creamos, más específicamente en la de “Datos Personales” colocando la misma información que el formulario anterior:

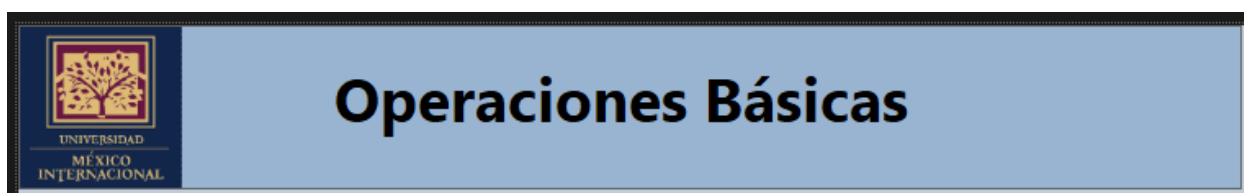
```
private void datosPersonalesToolStripMenuItem_Click(object sender, EventArgs e)
{
    Formularios.Form_Informacion_Personal Form_Informacion_Personal = new Formularios.Form_Informacion_Personal();
    Form_Informacion_Personal.Show();
    this.Hide();
}
```

El botón de regresar también le colocamos su codificación para que nos regrese a la ventana principal:

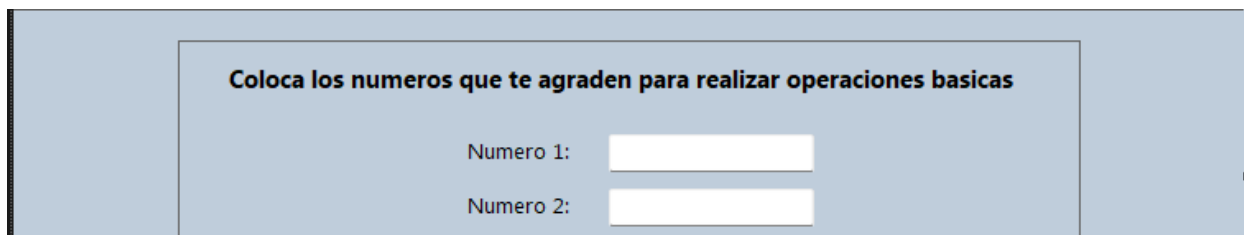
```
1 referencia
private void btn_Regresar_Click(object sender, EventArgs e)
{
    this.Close();
    Form_Menu back = new Form_Menu();
    back.Show();
}
```

Continuaremos con la siguiente ventana en la que se llamará “Operaciones Básicas” en ella como su nombre lo indica se podrá realizar sumas, restas, multiplicaciones y divisiones de manera práctica, pero antes de eso crearemos el formulario con su nombre respectivo:

 Form_Operaciones_Basicas.cs



Colocaremos etiquetas que digan, “Numero 1” y “Numero 2” a un lado de estas estarán cajas de texto para que el usuario coloque el número que el desee, además de una pequeña instrucción de lo que hace el programa:



Colocaremos botones que representen la suma, resta, multiplicación y división:



Una etiqueta que diga “Resultado” y a un lado una caja de texto para que indique el resultado de cada operación:



Agregaremos botones, uno de “Limpiar” y otro de “Regresar”, cada uno de ellos tendrá la acción correspondiente:



Es momento de crear una clase con el nombre que representa el formulario para evitar confundirnos:

C# Clase_Operaciones_Basicas.cs

Crearemos una variable de tipo “int”, tendremos números enteros solamente en la suma, resta y multiplicación, después de ello se crean las variables con las que se tomaran las operaciones para dichos números, uno para la suma:

```
1 referencia
public int Sumar (int numero1, int numero2)
{
    int resultado;
    resultado = numero1 + numero2;
    return resultado;
}
```

Para la resta:

```
1 referencia
public int Restar (int numero1, int numero2)
{
    int resultado;
    resultado = numero1 - numero2;
    return resultado;
}
```

Para la Multiplicación:

```
1 referencia
public int Multiplicar (int numero1, int numero2)
{
    int resultado;
    resultado = numero1 * numero2;
    return resultado;
}
```

En la División es lo mismo solo que aquí se utilizó la variable de tipo “float” para que el resultado tenga decimales y así sea más preciso:

```
1 referencia
public float Division (float numero1, float numero2)
{
    float resultado;
    resultado = numero1 / numero2;
    return resultado;
}
```

Tenemos que unir la clase con las acciones de cada botón, nos dirigimos al botón de sumar y damos doble clic para enlazarlo, podemos ver que se crea una variable nueva del tipo “int” llamada “resultado”, se busca la carpeta en donde esta guardada la clase y el nombre de la clase que se colocó, se crea una nueva instancia con el mismo nombre para después llamar la variable “resultado” que será igual al nombre de clase que se creó anteriormente y el método de “Sumar”, ya en este punto se tiene que convertir la variable de tipo “int” en texto que lo tomara del “TextBox” del “Numero1” y convertir también la otra “TextBox” en donde se tomara el otro número, por último la “TextBox” de “resultado” mandara a llamar lo que se colocó en la “TextBox” de “Numero “ y en la “TextBox” del “Numero 2” que se volverá a convertir en carácter para que logre colocar el resultado en la caja de texto:

```
1 referencia
private void btn_suma_Click(object sender, EventArgs e)
{
    int resultado;
    Clases.Clase_Operaciones_Basicas clase_Operaciones_Basicas = new Clases.Clase_Operaciones_Basicas();
    resultado = clase_Operaciones_Basicas.Sumar(int.Parse(txt_Numero1.Text), int.Parse(text_Numero2.Text));
    txt_Resultado.Text = resultado.ToString();
}
```

Hacemos lo mismo con las otras operaciones solo cambiando la acción dependiendo de lo que se necesite:

```
1 referencia
private void btn_resta_Click(object sender, EventArgs e)
{
    int resultado;
    Clases.Clase_Operaciones_Basicas clase_Operaciones_Basicas = new Clases.Clase_Operaciones_Basicas();
    resultado = clase_Operaciones_Basicas.Restar(int.Parse(txt_Numero1.Text), int.Parse(text_Numero2.Text));
    txt_Resultado.Text = resultado.ToString();
}
```

Multipliación:

```
1 referencia
private void btn_multiplicacion_Click(object sender, EventArgs e)
{
    int resultado;
    Clases.Clase_Operaciones_Basicas clase_Operaciones_Basicas = new Clases.Clase_Operaciones_Basicas();
    resultado = clase_Operaciones_Basicas.Multiplicar(int.Parse(txt_Numero1.Text), int.Parse(text_Numero2.Text));
    txt_Resultado.Text = resultado.ToString();
}
```

Y por último la División:

```
1 referencia
private void btn_Division_Click(object sender, EventArgs e)
{
    float resultado;
    Clases.Clase_Operaciones_Basicas clase_Operaciones_Basicas = new Clases.Clase_Operaciones_Basicas();
    resultado = clase_Operaciones_Basicas.Division(float.Parse(txt_Numero1.Text), float.Parse (text_Numero2.Text));
    txt_Resultado.Text = resultado.ToString();
}
```

La “TextBox” de Resultado la esconderemos para que no se pueda colocar nada y solo nos muestre el resultado:

```
1 referencia
public Form_Operaciones_Basicas()
{
    InitializeComponent();
    txt_Resultado.Enabled = false;
}
```

Le daremos la acción al botón de “regresar” para que nos de la acción de enviarnos al menú principal:

```
1 referencia
private void btn_Regresar_Click(object sender, EventArgs e)
{
    this.Close();
    Form_Menu back = new Form_Menu();
    back.Show();
}
```

Y para el botón de “Limpiar”:

```
1 referencia
private void btn_limpiar_Click(object sender, EventArgs e)
{
    txt_Numero1.Clear();
    txt_Numero2.Clear();
    txt_Resultado.Clear();
}
```


Unimos dando doble clic en el índice, la sección “Operaciones Básicas” nos mandara a la codificación en donde haremos lo mismo que estamos haciendo con las otras:

```
1 referencia
private void operacionesBásicasToolStripMenuItem_Click(object sender, EventArgs e)
{
    Formularios.Form_Operaciones_Basicas Form_Operaciones_Basicas = new Formularios.Form_Operaciones_Basicas();
    Form_Operaciones_Basicas.Show();
    this.Hide();
}
```

El último apartado de salir es idéntico, damos clic en el índice, seleccionamos “Salir” y nos abrirá la codificación, solo ponemos lo siguiente:

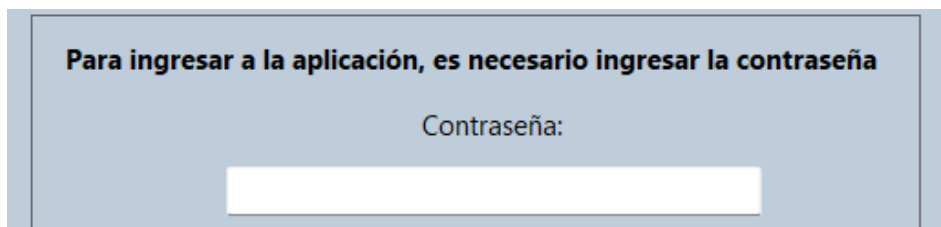
```
1 referencia
private void salirToolStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Falta lo más importante que es la ventana de inicio, para ello hacemos lo mismo, creamos un nuevo formulario en donde lo llamaremos “Ingreso” y realizamos el diseño de dicha ventana:

 Form_Ingreso.cs



Se coloca una instrucción para que sea más fácil de entender, después de ello una etiqueta que diga “Contraseña” y una “TextBox” donde se tenga que colocar la contraseña que nosotros decidiremos:



Un botón que diga “Entrar” y debajo de este una pequeña imagen de Bienvenida:



En este solamente damos clic en el botón de “Entrar” para que nos deje escribir el código, crearemos una sentencia, primero si el cuadro de texto de contraseña está vacío nos mandara un mensaje de que tenemos que ingresar la contraseña, ahora vamos a colocar una contraseña predefinida, si esto es correcto nos mandara al menú principal y la ventana de ingreso se esconderá para que sea más atractiva visualmente, caso contrario que si no se cumplen las dos primeras condiciones nos dirá que la contraseña es incorrecta y en automático se limpiara lo que escribimos en el cuadro de texto:

```
1 referencia
private void btn_Entrar_Click(object sender, EventArgs e)
{
    if (txt_Contraseña.Text == "")
    {
        MessageBox.Show("Favor de Ingresar una Contraseña.");
    }
    else if (txt_Contraseña.Text == "123456789")
    {
        Form_Menu form_Menu = new Form_Menu();
        form_Menu.Show();
        this.Hide();
    }
    else
    {
        MessageBox.Show("Contraseña Incorrecta.");
        txt_Contraseña.Clear();
    }
}
```

Para que la ventana de “Ingreso” aparezca antes que la de “Menú” tenemos que modificar el siguiente parámetro, se coloca el nombre de la ventana donde esta “Ingreso” y con esto el programa iniciara con la ventana de “Ingreso” para dar paso al de “Menú”:

```
Application.EnableVisualStyles();
Application.SetCompatibleTextRenderingDefault(false);
Application.Run(new Form_Ingreso());
```

Conclusión

El conocer lo básico en este tipo de lenguajes es fundamental para lograr tener un mejor entendimiento de lo que representa esta actividad, algo que puede resultar confuso o difícil, puede ser moldeado con practica y conocimiento, como ya se mencionó anteriormente, el realizar ejercicios para aprender más sobre algo, investigar, leer y sobre todo despejar dudas es muy importante, nos hace salir de nuestra zona de confort para adentrarnos a nuevos caminos y habilidades que nos permitan ser mejor en cada uno de los aspectos. En este caso crear un menú en vez de resultar tedioso fue interesante y a la vez entretenido, el crear un diseño que se vea acogedor y no arcaico fue una de las razones por las que me gustó mucho esa actividad y por supuesto hace que mi conocimiento en este tipo de tareas aumente cada día más. Es importante mencionar que el archivo del proyecto estará en el link de GitHub donde el maestro puede probar la actividad, más adelante se dará el link con el nombre de “Actividad 2”:

Link de GitHub

<https://github.com/Leyzu-Ing/Lenguaje-de-Programaci-n-III.git>

Referencias

Adegeo. (s. f.-b). *Tutorial: Proporcionar elementos de menú estándar a un formulario - Windows Forms .NET Framework*. Microsoft Learn. <https://learn.microsoft.com/es-es/dotnet/desktop/winforms/controls/walkthrough-providing-standard-menu-items-to-a-form?view=netframeworkdesktop-4.8>

Adegeo. (s. f.-a). *Controles que se utilizan - Windows Forms .NET Framework*. Microsoft Learn. <https://learn.microsoft.com/es-es/dotnet/desktop/winforms/controls/controls-to-use-on-windows-forms?view=netframeworkdesktop-4.8>