



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE TLAXIACO

PRACTICA 3: Visual Studio UX UI

Presenta:

No. Control 23620107 López García Lizeth Nallely

No. Control 23620286 Cortes Avila Alan Sahid

No. Control 23620254 Nicolas Gimenez Angel

No. Control 23620227 Ortiz Ortiz Abelardo

Carrera:

Ingeniería en Sistemas Computacionales

Cuarto Semestre

Asignatura:

Tópicos Avanzados de Programación

Encargado:

Jose Alfredo Roman Cruz

Tlaxiaco, Oaxaca a 18 de febrero de 2025

PRACTICA 2 (Visual Studio UI)

OBJETIVO

Desarrollar programas para interactuar con el usuario de una manera amigable, utilizando GUI (Interfaz Gráfica de Usuario) manipulables a través de eventos. El objetivo de crear una interfaz de escritorio permite mejorar la experiencia de los usuarios que interactúan con diferentes sistemas de control, monitorización o supervisión, al presentar la información de una manera visual e intuitiva con el entorno de Visual Studio.

MATERIAL

- Una laptop
- Software para el diseño de interfaces (Visual Studio)

PROCEDIMIENTO

1. Para que nuestras interfaces al ser ejecutadas estén centradas, damos clic en la interfaz, y en propiedades seleccionamos la opción “Size” y “Starposition” para cambiar la posición a “CenterScreen”. Lo mismo hacemos con todas las interfaces.

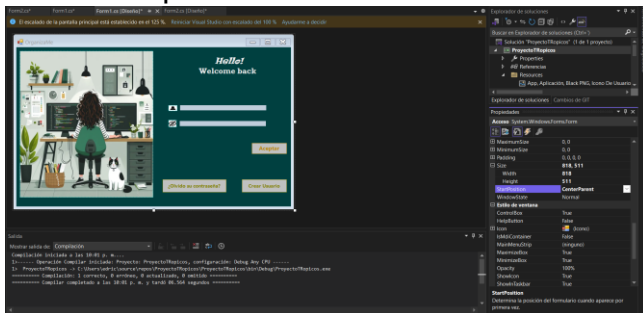


Figura 1

2. Cuando el usuario haya ingresado sus datos (usuario y contraseña), al dar clic en el botón de “aceptar”, pasará a la interfaz de inicio, por lo que en código ocupamos una función “llamar”.
3. Para que el usuario visualice las actividades de cada materia, dando doble clic al botón encontrado en la parte de materias, nos mandará al código del formulario, por lo que agregamos un código ocupando la función “Llamar” y llamamos al formulario de actividades. Lo mismo hacemos con todos los botones ubicados en el apartado de Materia.

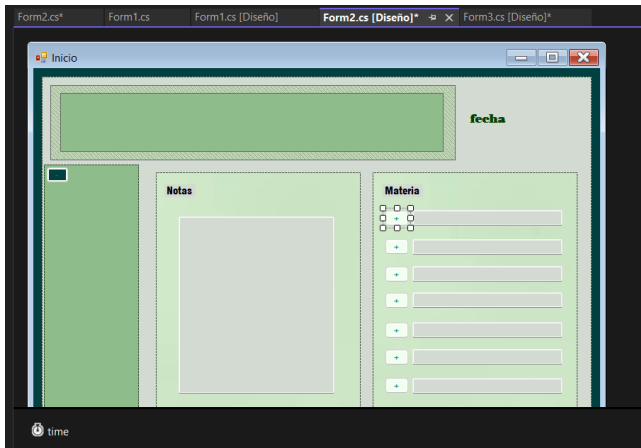


Figura 2

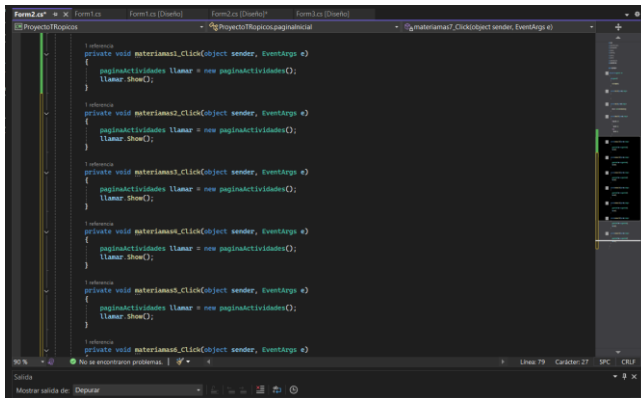


Figura 3

4. Para que el usuario ingrese datos correctos, creamos un código para que muestre cuadros de texto, los cuales ratificarán los datos y no halla equivocaciones a la hora de ingresar los datos.

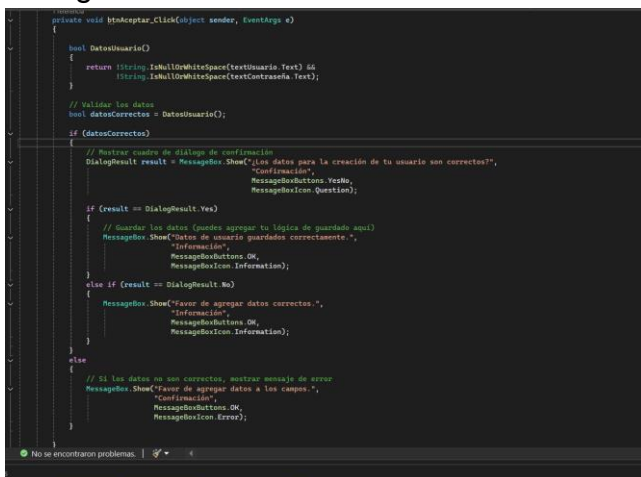


Figura 4

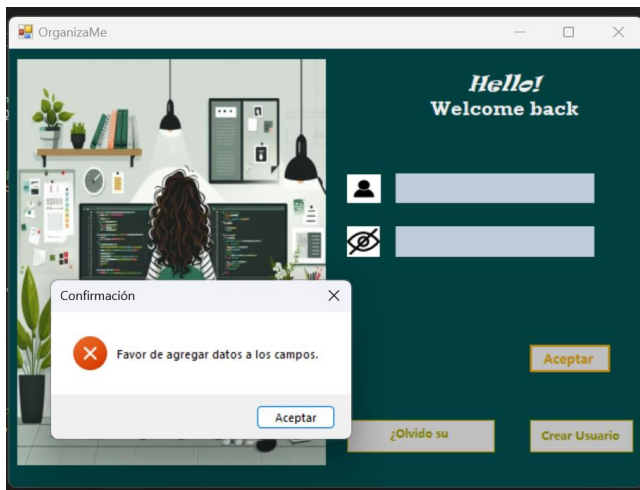


Figura 5

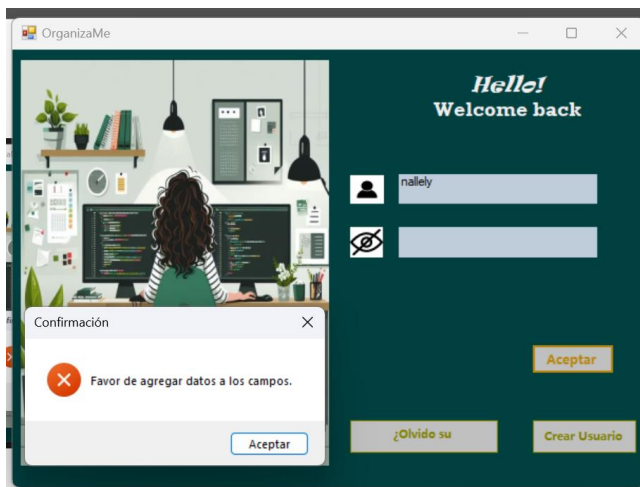


Figura 6

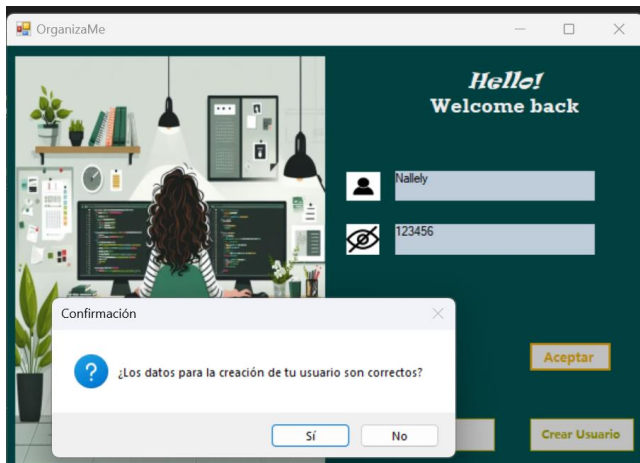


Figura 7

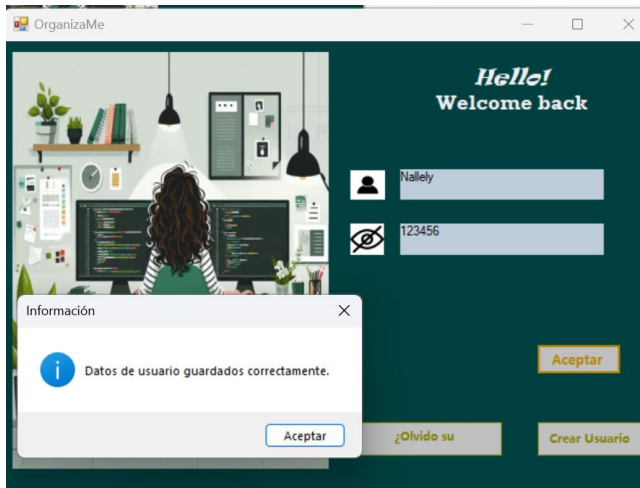


Figura 8

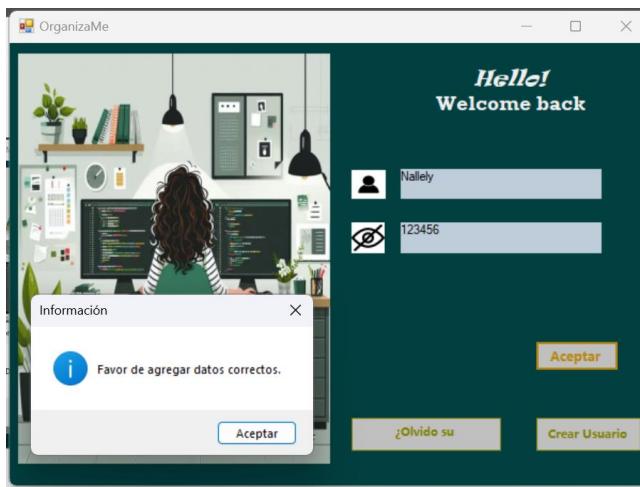


Figura 9

5. Para que los datos sean los correctos a la hora de crear un usuario, se implementa un código para que aparezcan cuadros de texto y se confirmen los datos.

```
private bool DatosUsuario()
{
    return !String.IsNullOrEmpty(textConfirmar.Text);
    return !String.IsNullOrEmpty(textUsuario.Text);
    return !String.IsNullOrEmpty(textCorreo.Text);
    return !String.IsNullOrEmpty(textContraseña.Text);
}

// referencia
private void btnIniciar_Click(object sender, EventArgs e)
{
    // Validar los datos (aquí puedes agregar tu lógica de validación)
    bool datosCorrectos = DatosUsuario();

    if (datosCorrectos)
    {
        // Muestra cuadro de diálogo de confirmación
        DialogResult result = MessageBox.Show("¿Los datos para la creación de tu usuario son correctos?", "Confirmación",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question);

        if (result == DialogResult.Yes)
        {
            // Guardar los datos (aquí puedes agregar tu lógica de guardado)
            MessageBox.Show("Datos de usuario guardados correctamente.",
                "Información",
                MessageBoxButtons.OK,
                MessageBoxIcon.Information);
        }
        if (result == DialogResult.No)
        {
            MessageBox.Show("Favor de agregar datos correctos.", "Información",
                MessageBoxButtons.OK,
                MessageBoxIcon.Information);
        }
    }
}
```

Figura 10

```

        MessageBox.Show("Favor de agregar datos correctos.", "Informacion",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        // Enfoque el TextBox
        textConfirmar.Focus();
        textContraseña.Focus();
        textCorreo.Focus();
        textConfirmar.Focus();
    }
}
else
{
    DialogResult result = MessageBox.Show("Favor de agregar datos a los campos.", "confirmacion",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
}

```

Figura 11

Al presionar el botón de iniciar sesión y no tenga datos en ningún campo aparece lo siguiente



Figura 12

Al agregar datos en los campos y después se pulsa el botón de iniciar sesión aparece el siguiente texto

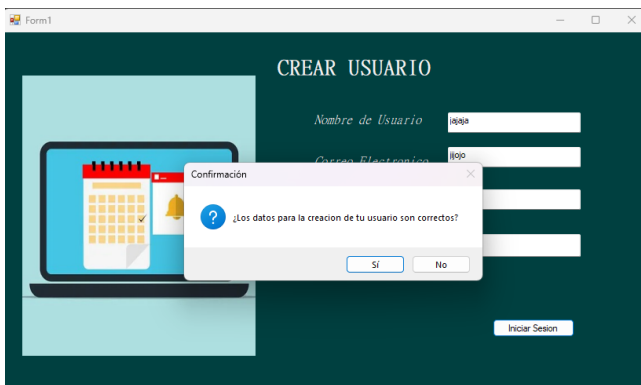


Figura 13

Y al pulsar el botón de SI aparece el siguiente texto de confirmacion

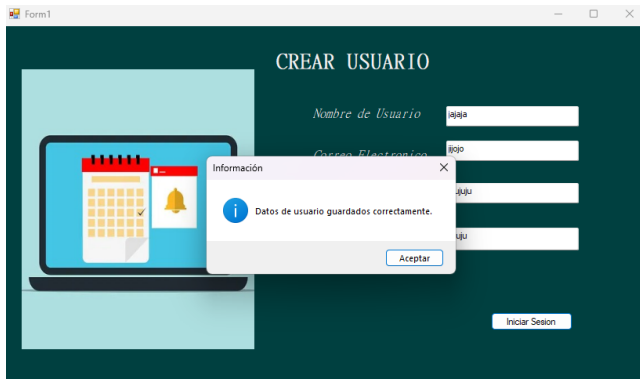


Figura 14

Y si se pulsa el boton de NO aparece el siguiente texto de petición

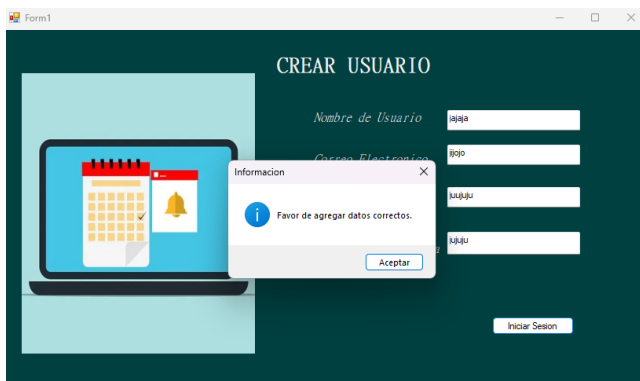


Figura 15

6. Para que los datos sean los correctos a la hora de cambiar contraseña, se implementa un código para que aparezcan cuadros de texto y se confirmen los datos.

Al entrar al apartado de cambiar la contraseña y no rellenar los campos solicitados aparecerá el siguiente mensaje

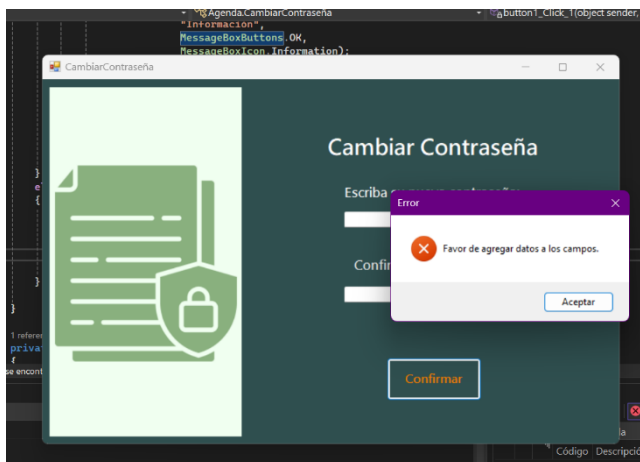
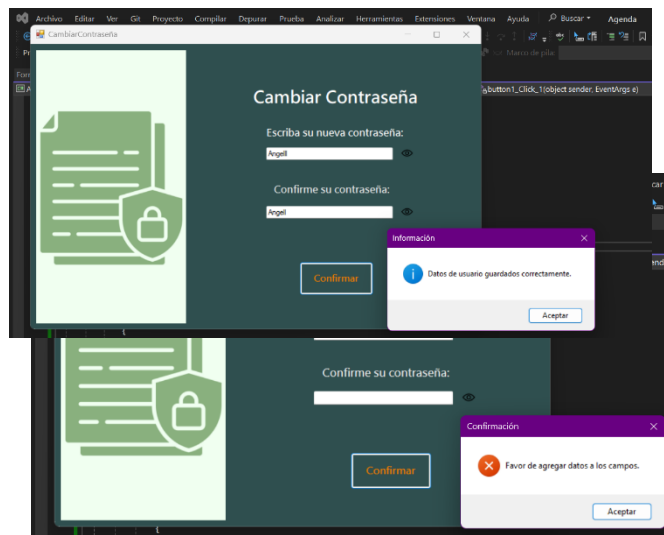


Figura 16



En este mismo campo al intentar cambiar la contraseña sin rellenar los dos recuadros aparece el siguiente mensaje

Figura 17

El siguiente paso es insertar los datos correctamente para así poder cambiar la contraseña

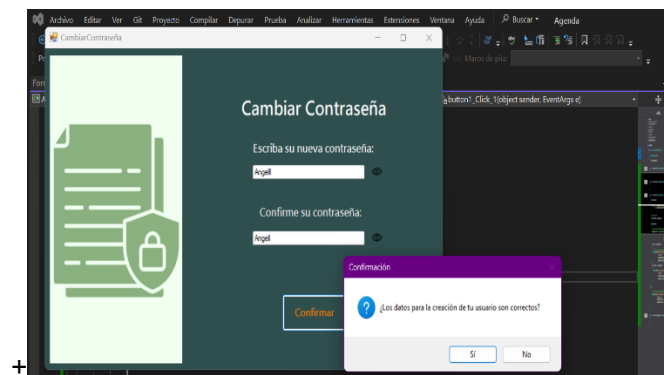


Figura 18

Una vez concluida aparecerá el mensaje mostrando que todo se ha hecho correctamente

AL agregar los datos incorrectos y marcar no en el mensaje aparecera lo siguiente

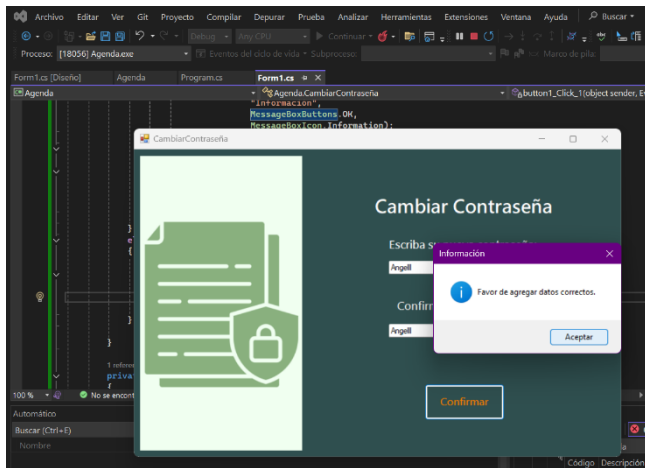


Figura 19

CONCLUSIONES

En esta práctica, se implementó lo que es el flujo de trabajo, tomando en cuenta todas las opciones y consideraciones que el usuario pueda tener, tomando en cuenta que si un usuario ingresa datos incorrectos, puede ratificar para no cometer ese tipo de errores, la simulación de este flujo de trabajo nos ayuda a comprender la parte del usuario para mejorar su experiencia en nuestra plataforma, aplicación o trabajo que nosotros como programadores vallamos a realizar. (Lizeth Nallely López García)

En esta práctica que realizamos que es la practica tres es escribir los eventos de las interfaces en lenguaje para que al dar clic en los botones aparezcan la ventana es lo que se realizó en esta práctica, ya que Visual Studio es un software que nos permite diseñar interfaces y escribir los eventos, configurar los botones, configurar colores, tipos de letras y combinar colores, iconografía y flujo de trabajo que nosotros queramos en nuestro diseño. La interfaz que nosotros realizamos es una interfaz clara, sencilla e intuitiva, permitiendo que los que ocupen nuestra aplicación puedan acceder rápidamente a las herramientas necesarias sin complicaciones. (Abelardo Ortiz Ortiz)

En esta practica se empezó a programar las funciones de las interfaces para la utilización del programa, se hizo la centralización de los formularios, así como para ponerle el maximizado de pantalla los campos que estén en la interfaz se acople a la pantalla para que no se desfigure el contenido de la interface y se pierda el atractivo de la aplicación por no acoplarse a la pantalla. (Alan Sahid Cortes Ávila)

Al llegar a esta parte del proyecto nos profundizamos mas en Visual Studio y el lenguaje de programación c#, aquí se empezó a interactuar mas con las funciones que lleva la aplicación que está en desarrollo, también se conoció un poco de lo mucho que es este lenguaje de programación, se hizo la aplicación mas interactiva para los usuarios que en un futuro la utilizarán para su bienestar, con estas practicas he aprendido mucho sobre lo que hace un mes era completamente desconocido para mi. (Angel Nicolas Gimenez)