

Actividad 2. Creación de Interfaces

Lenguajes de Programación III

Ingeniería en Desarrollo de Software

Tutor: Francisco Ortega Rivera

Alumno: José Luis Pacheco González

Fecha: 26 de noviembre 2023

Índice

Introducción	3
Descripción	4
Justificación	5
Desarrollo.....	6
• Entorno de trabajo	6
• Entorno del programa.....	12
Conclusión	15
Referencias.....	16

Introducción

El desarrollo de interfaces en Visual Studio para el proyecto "Kamil" resulta fundamental para poder administrar de manera eficiente las operaciones de Altas, Bajas y Cambios en la información de los empleados. Visual Studio, como un entorno integrado de desarrollo, proporciona un conjunto sólido de herramientas que simplifican la creación de interfaces de usuario intuitivas y prácticas.

En el caso de las Altas, la interfaz debe posibilitar la introducción eficaz de nuevos datos de empleados, asegurando la validación de la información ingresada para preservar la integridad de la base de datos. La implementación de formularios claros y la definición precisa de campos en Visual Studio facilita este proceso.

Por otro lado, para capturar las Bajas, se requiere una interfaz que permita la selección y confirmación de empleados a ser dados de baja. Un diseño meticuloso puede integrar medidas de seguridad para prevenir eliminaciones accidentales y confirmaciones redundantes.

En cuanto a los Cambios, la interfaz debe mostrar de manera efectiva los datos existentes y permitir su modificación. Para todo esto Visual Studio ofrece controles de edición y validación que aseguran una manipulación precisa de la información.

Descripción

Para el desarrollo de esta actividad es necesario recapitular en contexto principal del proyecto, el cual consiste en la creación de un sistema de gestión de empleados para la empresa "Kamil" a través de un proceso ABC. En esta parte del proyecto se requiere elaborar el diseño de interfaces utilizando Visual Studio, que incluirán un menú de opciones, una interfaz para el ingreso de datos de empleados, directivos y centros.

El trabajo presentado se basa en la programación orientada a objetos para gestionar de manera eficaz la información de los empleados y sus posibles cambios. El diseño de interfaces tiene como objetivo simplificar la interacción del usuario con el sistema, permitiendo una gestión intuitiva de altas, bajas y consultas. Como se mencionó anteriormente todo se llevará a cabo en Visual Studio, aprovechando sus capacidades de desarrollo de software. En última instancia, este sistema proporcionará a la empresa "Kamil" una herramienta integral para mejorar la administración de su personal, incrementando la organización y accesibilidad de la información relacionada con empleados, directivos y centros.

Justificación

Utilizar Visual Studio para el desarrollo de interfaces en un sistema de gestión de base de datos en SQL Server se fundamenta en varias razones. Visual Studio brinda un entorno integrado que simplifica la creación de interfaces de usuario intuitivas y eficaces, permitiendo a los desarrolladores diseñar aplicaciones con características avanzadas de manera rápida.

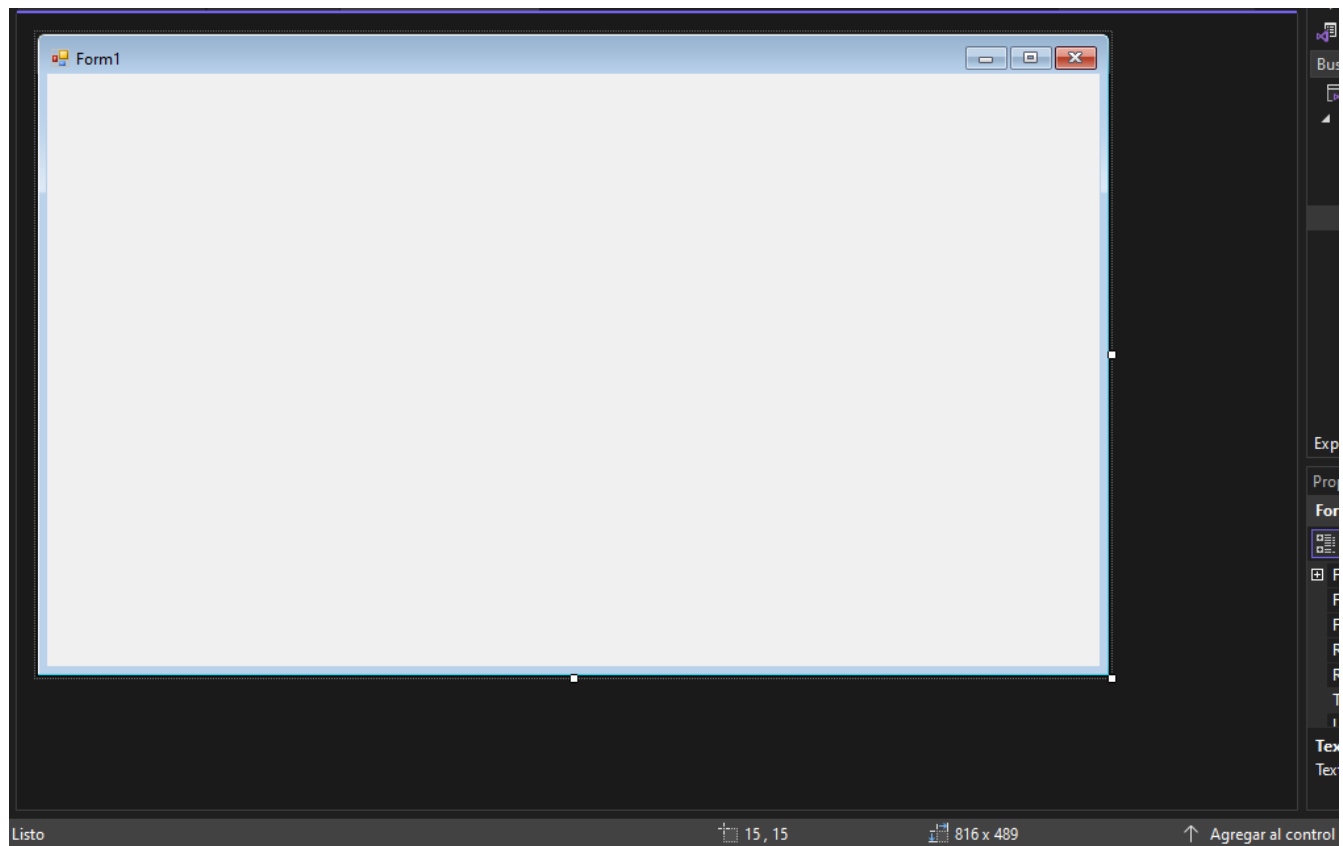
Al integrar Visual Studio con SQL Server se accede a un abanico de herramientas específicas para la manipulación de bases de datos, con las cuales se simplifica la interacción entre la interfaz y la gestión de datos. La creación de formularios y controles visuales en Visual Studio facilita la entrada y modificación de datos, lo cual mejora la experiencia del usuario y minimiza errores.

Esta integración promueve la coherencia y eficiencia en el desarrollo, ya que los cambios en la base de datos se reflejan directamente en la interfaz y viceversa. Esto simplifica el mantenimiento y la evolución del sistema.

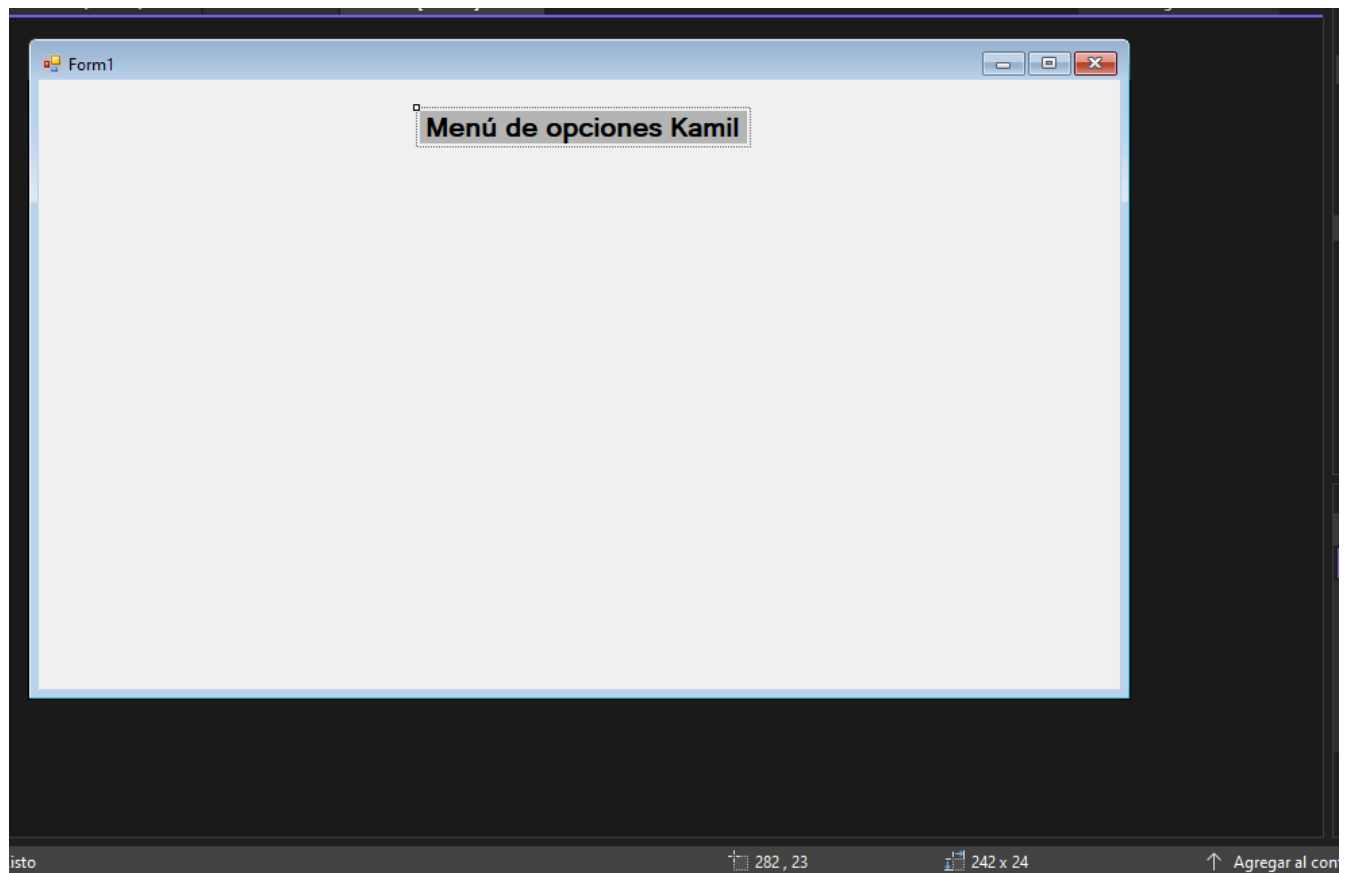
Desarrollo

- **Entorno de trabajo**

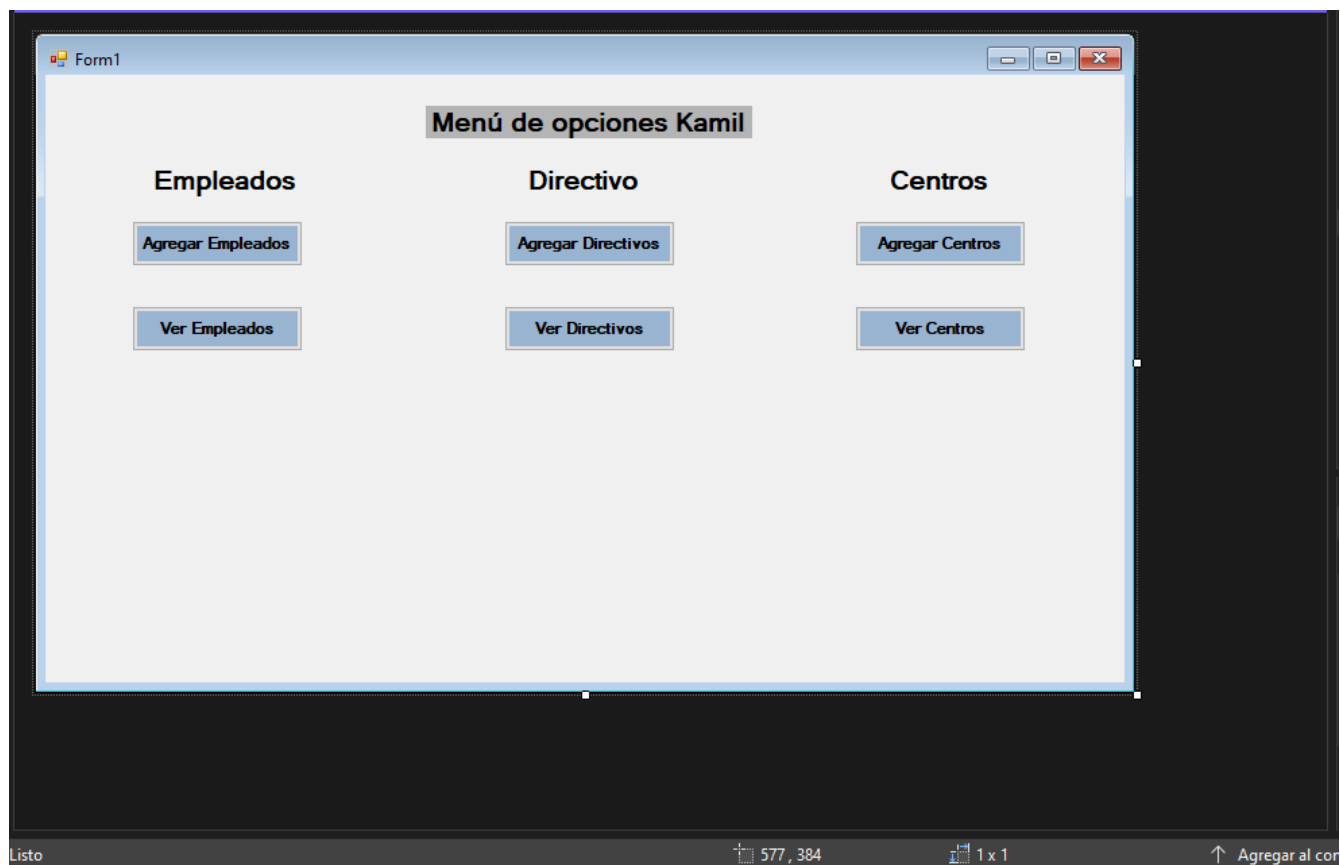
En las siguientes capturas se muestra la evidencia de la creación de formularios en Visual Studio.



Aquí se puede apreciar el avance el menú de opciones.



En esta captura se muestra terminada la interfaz principal, que corresponde al menú de opciones, en base a esto se crearon los demás formularios correspondientes a Empleados, Directivos y Centros, las capturas se muestran a continuación.



Form4.cs **Form4.cs [Diseño]** Form3.cs [Diseño] Form2.cs Form2.cs [Diseño] Form1.cs Form1.cs [Diseño] << >> ⚙

Formulario empleados

Número:

Nombre:

Apellido paterno:

Apellido materno:

Fecha de nacimiento:

Buscar

RFC:

Centro de trabajo:

Puesto:

Directivo:

Guardar

Modificar

Eliminar

Form3.cs [Diseño] Form2.cs Form2.cs [Diseño] Form1.cs Form1.cs [Diseño] Program.cs

Formulario Directivos

Número: Buscar

Centro supervisado:

Ciudad:

Número empleado: Guardar Modificar Eliminar

Lista de errores

Form2.cs **Form2.cs [Diseño]** Form1.cs Form1.cs [Diseño] Program.cs

Formulario de centros

Número:

Nombre:

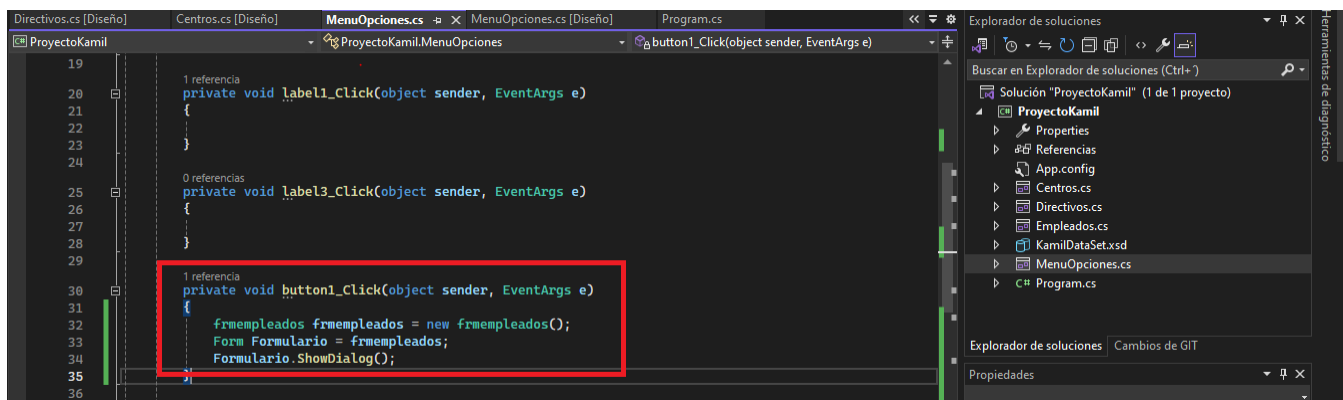
Ciudad:

Lista de errores

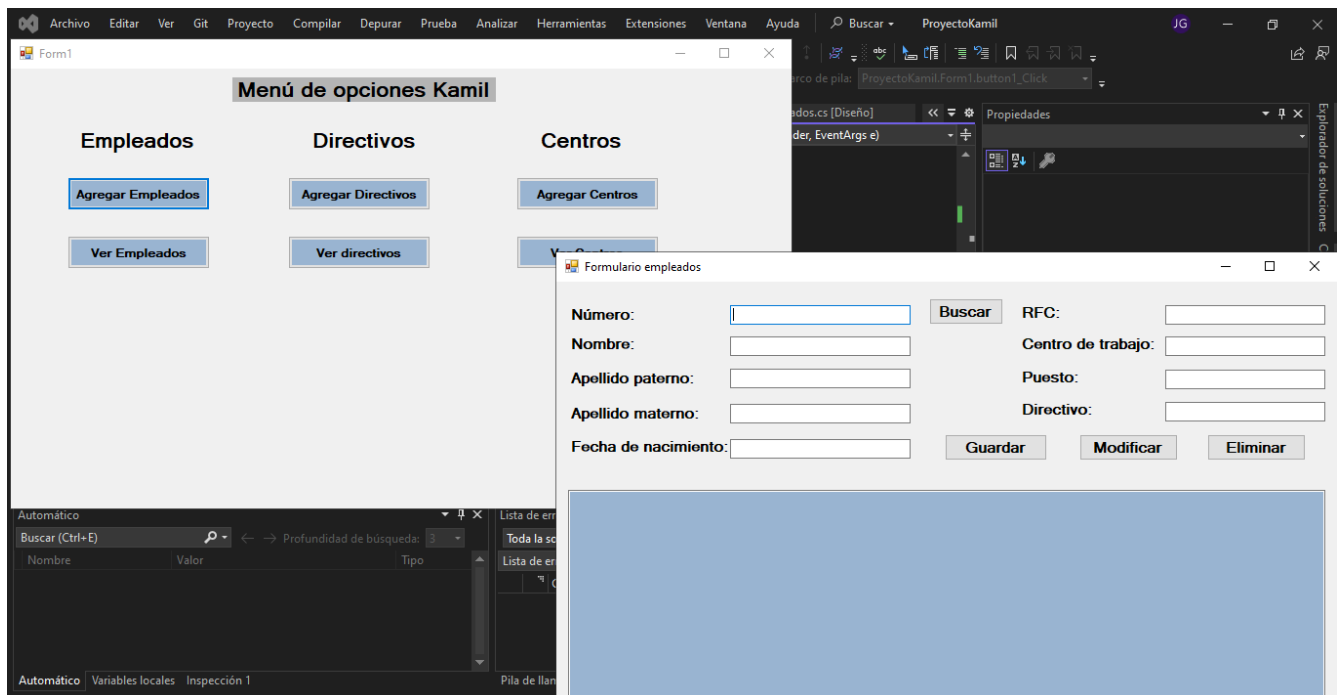
- **Entorno del programa**

En esta parte del proyecto se muestra el código para declarar la variable que realiza la función de llamar al formulario que corresponda a la opción seleccionada en el menú.

De igual manera se adjuntan las capturas del programa funcionando, con las diferentes opciones del menú siendo ejecutadas, llamando al formulario correcto.



```
19
20
21 1 referencia
22 private void label1_Click(object sender, EventArgs e)
23 {
24 }
25
26 0 referencias
27 private void label3_Click(object sender, EventArgs e)
28 {
29 }
30
31 1 referencia
32 private void button1_Click(object sender, EventArgs e)
33 {
34     frmempleados frmempleados = new frmempleados();
35     Form Formulario = frmempleados;
36     Formulario.ShowDialog();
37 }
```



Menú de opciones Kamil

Empleados	Directivos	Centros
<button>Agregar Empleados</button> <button>Ver Empleados</button>	<button>Agregar Directivos</button> <button>Ver directivos</button>	<button>Agregar Centros</button> <button>Ver Centros</button>

Formulario empleados

Número:	<input type="text"/>	Buscar	RFC:	<input type="text"/>
Nombre:	<input type="text"/>		Centro de trabajo:	<input type="text"/>
Apellido paterno:	<input type="text"/>		Puesto:	<input type="text"/>
Apellido materno:	<input type="text"/>		Directivo:	<input type="text"/>
Fecha de nacimiento:	<input type="text"/>	Guardar	Modificar	Eliminar

```
Directivos.cs [Diseño]  Centros.cs [Diseño]  MenuOpciones.cs  MenuOpciones.cs [Diseño]  Program.cs
ProyectoKamil
36
37 1 referencia
38 private void Form1_Load(object sender, EventArgs e)
39 {
40
41
42 1 referencia
43 private void button2_Click(object sender, EventArgs e)
44 {
45
46
47 1 referencia
48 private void button4_Click(object sender, EventArgs e)
49 {
50     frmdirectivos frmdirectivos = new frmdirectivos();
51     Form Formulario = frmdirectivos;
52     Formulario.ShowDialog();
53 }
54
```

Explorador de soluciones

Buscar en Explorador de soluciones (Ctrl+)

Solución "ProyectoKamil" (1 de 1 proyecto)

- ProyectoKamil
 - Properties
 - Referencias
 - App.config
 - Centros.cs
 - Directivos.cs
 - Empleados.cs
 - KamilDataSet.xsd
 - MenuOpciones.cs
 - Program.cs

Explorador de soluciones Cambios de GIT

Propiedades

Archivo Editar Ver Git Proyecto Compilar Depurar Prueba Analizar Herramientas Extensiones Ventana Ayuda

ProyectoKamil

Form1

Menú de opciones Kamil

Empleados	Directivos	Centros
<button>Agregar Empleados</button>	<button>Agregar Directivos</button>	<button>Agregar Centros</button>
<button>Ver Empleados</button>	<button>Ver directivos</button>	<button>Ver Centros</button>

Formulario Directivos

Número: Buscar

Centro supervisado:

Ciudad:

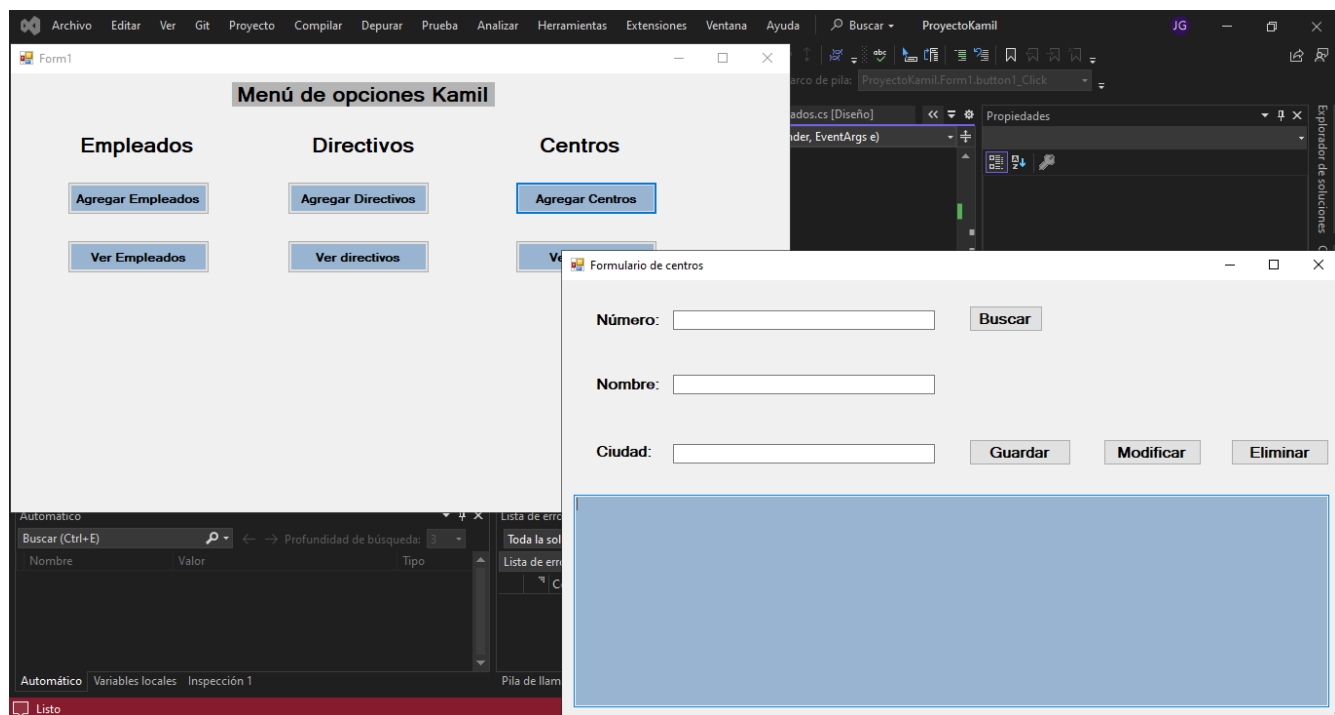
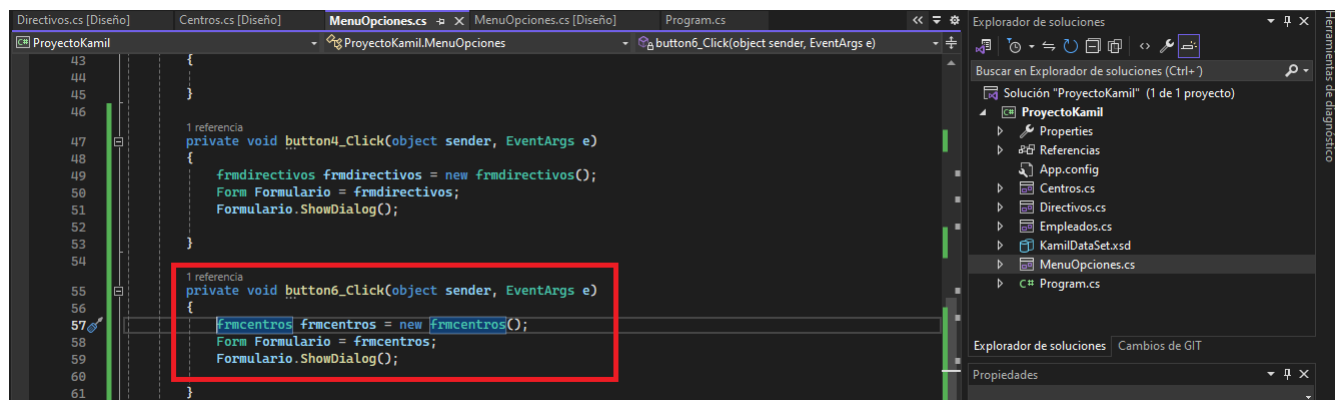
Número empleado: Guardar Modificar Eliminar

Buscar (Ctrl+E)

Nombre	Valor	Tipo
--------	-------	------

Automático Variables locales Inspección 1

Pila de llamadas



Conclusión

La implementación de formularios, menús y consultas en el ámbito de bases de datos resulta fundamental para aumentar la eficacia y la facilidad de uso en la administración de la información. Los formularios ofrecen una interfaz amigable que simplifica la introducción estructurada de datos, agilizando el proceso y minimizando posibles errores. Por su parte, los menús optimizan la navegación en la base de datos, mejorando la accesibilidad a la información almacenada.

Adicionalmente, las consultas juegan un papel esencial al posibilitar la extracción selectiva de datos basada en criterios específicos. Esto no solo perfecciona la recuperación de información, sino que también contribuye a la toma de decisiones informada. En conjunto estos componentes proporcionan una experiencia completa para los usuarios, aumentando la eficiencia operativa y la calidad en la gestión de datos.

Por último, la utilización eficaz de formularios, menús y consultas en bases de datos no solo simplifica las tareas cotidianas, sino que también aporta a la integridad y utilidad de la información almacenada, mejorando de manera general la funcionalidad y la eficacia del sistema.

Referencias

Amaury, D. [@DiogenesAmauryMartinez]. (2016, June 14). *Crear Menú Y Llamar Formularios en C#*. Youtube. https://www.youtube.com/watch?v=zG06vtko_mM

Veliz, J. [@joseveliz]. (2022, February 25). *9-COMO MOSTRAR UN FORMULARIO DESDE OTRO EN C# 2022 (MENUSTRIP-SHOW-SHOWDIALOG)*. Youtube.
https://www.youtube.com/watch?v=by_ihlGMGDM



Enlace GitHub