



Actividad | #2 |

Creación de Interfaces

Lenguajes de Programación III

Ingeniería en Desarrollo de Software



TUTOR: Miguel Ángel Rodríguez Vega

ALUMNO: Oscar Esteban Sánchez Leyva

FECHA: 27/Noviembre/2025

ÍNDICE

ÍNDICE	2
DESARROLLO	3
CREACIÓN DE INTERFACES.....	3
ESTRUCTURA DEL CÓDIGO C# Y LÓGICA CLAVE	6
REFERENCIAS.....	10

DESARROLLO

CREACIÓN DE INTERFACES

La creación de interfaces consiste en desarrollar tanto la apariencia visual como la interacción entre un usuario y un sistema tecnológico, ya sea un software o un dispositivo. Su propósito principal es garantizar una experiencia intuitiva, funcional y visualmente agradable, dando prioridad a la usabilidad. Aspectos esenciales como el diseño de la disposición, la tipografía, los colores y los elementos de navegación desempeñan un papel clave en el proceso.

Aspectos fundamentales en el diseño de interfaces:

- **Estética y usabilidad:** Es esencial encontrar el equilibrio entre un diseño atractivo y una funcionalidad intuitiva. Aunque la estética aporta valor, la usabilidad debe prevalecer, asegurando que el producto resulte sencillo de usar sin importar su nivel de diseño visual.
- **Diseño visual:** Aborda la elección de colores, tipografías, iconos y la organización de los componentes en pantalla. Por ejemplo, los colores pueden influir significativamente en las emociones del usuario; el azul puede evocar confianza, mientras que el rojo podría generar urgencia.
- **Interacción y navegación:** Se enfoca en cómo el usuario interactúa con el sistema, a través de elementos como botones, formularios y sistemas de navegación que conecten distintas secciones. Las acciones deben ser claras y los controles organizados de forma lógica para facilitar su uso.

- **Coherencia:** Mantener una experiencia uniforme y alineada con la identidad de la marca en todas las partes del sistema es esencial.
- **Adaptabilidad:** Es fundamental que el diseño sea capaz de ajustarse eficientemente a distintos dispositivos y plataformas, como móviles, escritorios o tabletas.

Tipos de interfaces:

- **Interfaces gráficas de usuario (GUI):** Son las más comunes y permiten la interacción mediante elementos visuales como menús, iconos y ventanas. El escritorio de Windows es un ejemplo representativo.
- **Interfaces controladas por voz (VUI):** En estas, los usuarios se comunican mediante comandos de voz. Siri de Apple y Alexa de Amazon son ejemplos populares.
- **Interfaces basadas en gestos:** Permiten la interacción a través de movimientos corporales, como sucede en los sistemas de realidad virtual.

Proceso de creación:

- 1. Investigación y planificación:** Se establecen los objetivos principales, se analiza y comprende al público objetivo, y se investigan las tendencias actuales en diseño para garantizar relevancia e innovación.
- 2. Diseño (wireframing y prototipado):** Se generan esquemas iniciales, a menudo a mano o con herramientas digitales, para definir la estructura de la interfaz. Posteriormente, se elaboran prototipos interactivos que pueden probarse directamente en un navegador.
- 3. Diseño detallado:** Se desarrollan los aspectos visuales definitivos, como la paleta de colores, la tipografía y otros elementos gráficos, utilizando software especializado.
- 4. Implementación:** Se lleva a cabo la programación necesaria para transformar las ideas en una interfaz funcional y operativa.
- 5. Pruebas y mejora:** Se procesan evaluaciones con usuarios reales para detectar posibles inconvenientes y adaptar el diseño de manera iterativa, optimizando continuamente la experiencia final.

ESTRUCTURA DEL CÓDIGO C# Y LÓGICA CLAVE

Para llevar a cabo esta propuesta, se generarán cinco formularios principales en una aplicación Windows Forms desarrollada con C#. Estos formularios cubrirán diferentes funcionalidades y estarán diseñados siguiendo la lógica clave descrita a continuación.

1. Formulario 1 (“frmIngreso”) Pantalla de acceso con contraseña:

Este formulario será el primero en ejecutarse y su propósito principal es validar la contraseña ingresada para permitir el acceso al formulario principal (“frmMenu”).

Controles necesarios:

- (“txtContraseña”): TextBox para ingresar la contraseña.
- (“btnEntrar”): Botón que verifica la contraseña e inicia sesión si es correcta.

Lógica principal:

- Configurar este formulario como el inicial en el archivo (“Program.cs”).
- Al presionar el botón (Entrar), se validará la contraseña comparándola con un valor predefinido (por ejemplo, "admin").
- Si la contraseña es correcta, se cerrará este formulario y se abrirá el formulario (“frmMenu”); de lo contrario, se mostrará un mensaje de error.

2. Formulario 2 (“frmMenu”), formulario principal con (“MenuStrip”):

Se trata de la interfaz principal de la aplicación, donde se gestionará la navegación entre las diferentes funcionalidades del programa.

Controles necesarios:

(“MenuStrip”): Con las siguientes opciones:

- Saludo: Abre el formulario (“frmSaludo”).
- Datos personales: Abre el formulario (“frmDatosPersonales”).
- Operaciones básicas: Abre el formulario (“frmOperacionesBasicas”).
- Salir: Finaliza la aplicación.

Lógica principal:

- Cada opción del menú abrirá el formulario correspondiente.
- Al usar "Salir", se cerrará toda la aplicación.

3. Formulario 3 (“frmSaludo”), saludo personalizado:

El propósito de este formulario es capturar un nombre mediante un cuadro de texto y mostrar un mensaje personalizado.

Controles necesarios:

- (“txtNombre”): TextBox para ingresar el nombre.
- (“btnSaludar”): Botón para generar el saludo en un (“MessageBox”).
- (“btnRegresar”): Botón para volver al formulario principal (“frmMenu”).

Lógica principal:

- Al hacer clic en el botón `Saludar`, se mostrará un mensaje con el formato “¡Hola, [nombre ingresado]!”, siempre verificando que no esté vacío.

4. Formulario 4 (“frmOperacionesBasicas”), cálculo de operaciones

matemáticas:

Este formulario permite realizar cálculos básicos con hasta seis números ingresados por el usuario. También debe incluir funcionalidades para limpiar los campos una vez que haya completado la operación.

Controles necesarios:

- Seis (“TextBox”) para capturar los números (“txtNum1” a “txtNum6”).
- Un (“TextBox”) para mostrar resultados (“txtResultado”).
- Botones específicos para cada función matemática requerida, como suma, resta, multiplicación, promedio, etc.

Lógica clave:

- Crear una función para leer y validar las entradas de los (“TextBox”). Esta función debe verificar que todos contengan valores numéricos válidos (tipo “double”) antes de realizar los cálculos.
- Implementar métodos separados para cada operación matemática (suma, resta, etc.), asegurándose de mostrar los resultados en el cuadro correspondiente (“txtResultado”).
- Incluir un botón adicional para limpiar los campos (“TextBox”) al terminar.

5. Formulario 5 (“frmDatosPersonales”), captura de información personal:

Este formulario estará dividido en pestañas utilizando un control (“TabControl”), permitiendo capturar información personal y clínica del usuario. Las dos pestañas serán: *Datos personales* y *datos clínicos*.

Controles necesarios:

- (“tabControl1”): Control con dos pestañas (“tabPageDatosPersonales” y “tabPageDatosClinicos”).
- En *Datos Personales*:
 1. (“dateTimePicker1”): Selector para la fecha de nacimiento.
 2. (“cmbEstadoCivil”): ComboBox para elegir entre opciones como soltero, casado, divorciado, etc.
- (“rbtHombre”) y (“rbtMujer”): (“RadioButtons”) para seleccionar el género.
- En *Datos Clínicos*: Agregar campos adicionales según necesidad específica.

Lógica clave:

- Inicializar las opciones del ComboBox (“cmbEstadoCivil”) dentro del evento Load del formulario (“Form_Load”). Ejemplo: agregar dinámicamente las opciones "Soltero".

REFERENCIAS

- Mikejo. (s. f.). *Implementación de una interfaz - Visual Studio (Windows)*. Microsoft Learn. <https://learn.microsoft.com/es-es/visualstudio/ide/reference/implement-interface?view=visualstudio>
- Visual Studio IDE con .NET: desarrolle cualquier aplicación con C#, F#, VB.* (2024, 12 diciembre). Visual Studio. <https://visualstudio.microsoft.com/es/vs/features/net-development/>
- BillWagner. (s. f.). *Creación e implementación de interfaces - Visual Basic*. Microsoft Learn. <https://learn.microsoft.com/es-es/dotnet/visual-basic/programming-guide/language-features/interfaces/walkthrough-creating-and-implementing-interfaces>
- W3Schools.com. (s. f.). https://www.w3schools.com/cs/cs_interface.php
- BillWagner. (s. f.-b). *Interfaces - Visual basic*. Microsoft Learn. <https://learn.microsoft.com/es-es/dotnet/visual-basic/programming-guide/language-features/interfaces/>