

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Materia:

Tópicos Avanzados de Programación.

Semestre:

4° Semestre.

Producto Académico:

Reporte de Práctica: “Práctica 3 U1 Cap. 3”.

Presenta(n):

Alejandro Zurita Pérez

Docente:

M.T.I. Dionisio Pérez Pérez.



Medellín de Bravo, Ver. SEP. – DIC. 20



VERACRUZ
GOBIERNO
DEL ESTADO

1. INTRODUCCION

Los ejercicios del 1 al 5 con Windows Forms son una oportunidad para explorar y practicar el desarrollo de aplicaciones de escritorio utilizando el framework .NET y el entorno de desarrollo integrado (IDE) de Visual Studio. Estos ejercicios cubren una variedad de conceptos básicos de programación y te familiarizarán con la creación de interfaces gráficas de usuario (GUI) interactivas y funcionales.

Ejercicio 1: Determinar si un número es par o impar

En este ejercicio, crearás una aplicación que permita al usuario ingresar un número y determinar si es par o impar. Utilizarás controles de cajas de texto y botones para capturar la entrada del usuario y mostrar el resultado de la verificación.

Ejercicio 2: Conversión entre grados y radianes

En este ejercicio, construirás una aplicación para convertir entre grados y radianes. El usuario podrá seleccionar la conversión deseada y proporcionar el valor a convertir. Utilizarás controles de lista desplegable y cajas de texto para capturar la entrada del usuario y mostrar los resultados de las conversiones.

Ejercicio 3: Cálculo de impuestos para productos

En este ejercicio, desarrollarás una aplicación para calcular impuestos sobre productos basados en la selección del usuario. Si el producto es una medicina, se aplicará un impuesto del 0%; de lo contrario, se aplicará un impuesto del 16%. Utilizarás controles de lista desplegable y cajas de texto para capturar la entrada del usuario y mostrar los resultados de los cálculos.

Ejercicio 4: Asignar nombre a un día de la semana según un número

En este ejercicio, crearás una aplicación que permita al usuario ingresar un número del 1 al 7 y mostrará el nombre correspondiente del día de la semana. Utilizarás un control de lista desplegable para seleccionar el número del día y un control de etiqueta para mostrar el nombre del día.

Ejercicio 5: Cálculo del perímetro y área de un polígono regular

En este ejercicio, construirás una aplicación para calcular el perímetro y el área de un polígono regular. El usuario podrá seleccionar el cálculo deseado y proporcionar los datos necesarios, como el número de lados y la longitud del lado. Utilizarás controles de lista desplegable, cajas de texto y etiquetas para capturar la entrada del usuario y mostrar los resultados de los cálculos.

Estos ejercicios proporcionan una base sólida para el desarrollo de aplicaciones de Windows Forms y te ayudarán a desarrollar habilidades esenciales en la programación en C# y .NET.

2. OBJETIVOS (COMPETENCIAS)

Es esta práctica se alcanzarán las siguientes competencias:

- Desarrollo de habilidades en programación en C#: Los ejercicios proporcionan la oportunidad de aplicar conceptos fundamentales de programación en C# como variables, estructuras de control, métodos, manejo de eventos, y más.
- Comprensión de la programación orientada a eventos: Al trabajar con controles de formularios y eventos en Windows Forms, se desarrolla la comprensión de cómo responder a las interacciones del usuario y ejecutar acciones correspondientes.
- Aprendizaje del desarrollo de aplicaciones de escritorio: Los ejercicios ayudan a familiarizarse con el desarrollo de aplicaciones de escritorio mediante el uso del framework .NET y el entorno de desarrollo integrado (IDE) de Visual Studio.
- Dominio de la creación de interfaces gráficas de usuario (GUI): Se adquieren habilidades en la creación de interfaces de usuario atractivas y funcionales utilizando controles como botones, cajas de texto, etiquetas, listas desplegables, y más.
- Práctica en el diseño de aplicaciones interactivas: Al permitir al usuario interactuar con la aplicación a través de la entrada de datos y la visualización de resultados, se practica el diseño de aplicaciones interactivas y amigables para el usuario.
- Aplicación de conceptos matemáticos y lógicos: En particular, los ejercicios 1, 2 y 5 implican el uso de conceptos matemáticos y lógicos, como determinar si un número es par o impar, realizar conversiones de unidades, y calcular perímetros y áreas de polígonos.
- Desarrollo de habilidades de resolución de problemas: Al enfrentarse a desafíos específicos en cada ejercicio, se desarrollan habilidades para analizar problemas, identificar soluciones y aplicarlas de manera efectiva en el desarrollo de software.
- Mejora de la capacidad de depuración y prueba de software: Durante el desarrollo de los ejercicios, se practica la depuración de errores y la realización de pruebas para garantizar el correcto funcionamiento de la aplicación.

Al lograr estos objetivos, los participantes estarán mejor preparados para abordar proyectos más avanzados de desarrollo de software y continuar su aprendizaje en el campo de la programación de aplicaciones de escritorio con C# y .NET.

3. FUNDAMENTO

Programación estructurada: Se practican conceptos de programación estructurada como variables, tipos de datos, operadores, estructuras de control (if, else, switch-case, while), y funciones (métodos en C#).

Eventos y manejo de eventos: Se aprende sobre el concepto de eventos y cómo manejarlos en una aplicación Windows Forms, permitiendo que la aplicación responda a las acciones del usuario, como hacer clic en un botón o seleccionar un elemento de una lista desplegable.

Interfaz de usuario (UI): Se desarrolla la comprensión de cómo crear interfaces de usuario efectivas y atractivas utilizando controles como botones, etiquetas, cuadros de texto, listas desplegables y más, y cómo organizarlos en un formulario de Windows.

Entrada y salida de datos: Se practica la entrada y salida de datos a través de controles de formulario, como cuadros de texto y listas desplegables, y cómo procesar esta información para realizar cálculos o mostrar resultados.

Manejo de errores y validación de entrada: Se abordan técnicas para manejar errores y validar la entrada del usuario, como asegurarse de que los datos ingresados sean del tipo y rango esperados, y mostrar mensajes de error descriptivos cuando sea necesario.

Algoritmos y cálculos matemáticos: Se aplican conceptos matemáticos y algoritmos para resolver problemas específicos, como determinar si un número es par o impar, realizar conversiones de unidades, y calcular perímetros y áreas de polígonos.

Estructuras de datos: Se utilizan estructuras de datos simples, como variables y arreglos, para almacenar y manipular información relevante para el problema a resolver.

Pruebas y depuración: Se practican habilidades de depuración para identificar y corregir errores en el código, así como realizar pruebas para garantizar que la aplicación funcione correctamente en diferentes situaciones y con diferentes tipos de entrada.

Dominar estos fundamentos es esencial para convertirse en un desarrollador de software competente y poder abordar proyectos más complejos y desafiantes en el futuro.

4. EQUIPAMIENTO Y MATERIAL DE APOYO

Computadora o Portátil: Se requiere una computadora o portátil funcional con un entorno de desarrollo instalado para practicar la escritura y ejecución de código en C#. Se recomienda utilizar un entorno de desarrollo integrado (IDE) como Visual Studio Community 2022 que proporcionan herramientas avanzadas para la programación en C#.

Acceso a Internet: El acceso a Internet es útil para acceder a recursos en línea, como documentación oficial de C#, tutoriales, foros de discusión y plataformas de aprendizaje en línea. Estos recursos pueden ayudar a los estudiantes a profundizar su comprensión de los conceptos y resolver problemas específicos durante las prácticas de programación.

Libros y Recursos de Aprendizaje: Se recomienda utilizar libros de texto y otros recursos de aprendizaje dedicados a la programación en C#. Estos recursos proporcionan una guía estructurada para aprender los fundamentos del lenguaje, así como ejemplos de código y ejercicios prácticos para reforzar el aprendizaje.

Comunidad de Desarrolladores: Unirse a comunidades en línea de desarrolladores de C# puede ser beneficioso para compartir conocimientos, hacer preguntas, obtener ayuda con problemas de codificación y colaborar en proyectos. Plataformas como Stack Overflow, Reddit (r/csharp), y foros de desarrolladores de Microsoft son excelentes lugares para interactuar con otros programadores de C#.

Práctica y Ejercicios: La práctica regular es fundamental para mejorar las habilidades de programación en C#. Los estudiantes deben dedicar tiempo a resolver problemas y completar ejercicios de programación para reforzar los conceptos aprendidos y desarrollar su capacidad para resolver problemas de manera independiente.

Al disponer del equipamiento y material de apoyo adecuados, se puede maximizar el aprendizaje y dominar los fundamentos de la programación en C#. Es importante dedicar tiempo y esfuerzo a practicar regularmente y explorar una variedad de recursos para obtener una comprensión completa del lenguaje y sus aplicaciones.

5. DESARROLLO DE LA PRACTICA

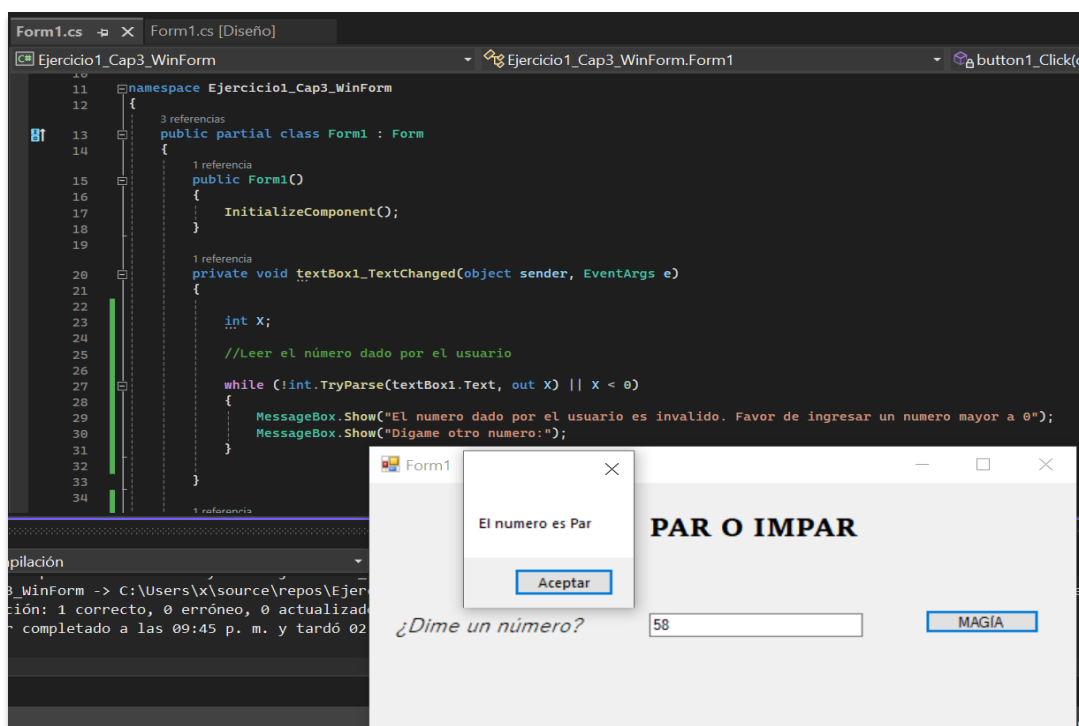
Ejercicio 1: Determinar si un número es par o impar

1. Interfaz de usuario:

- Un cuadro de texto para que el usuario ingrese el número.
- Un botón para que el usuario solicite la verificación.
- Una etiqueta para mostrar el resultado (par o impar).

2. Funcionalidad:

- Cuando el usuario hace clic en el botón, se verifica si el número ingresado es par o impar.
- El resultado se muestra en la etiqueta correspondiente.



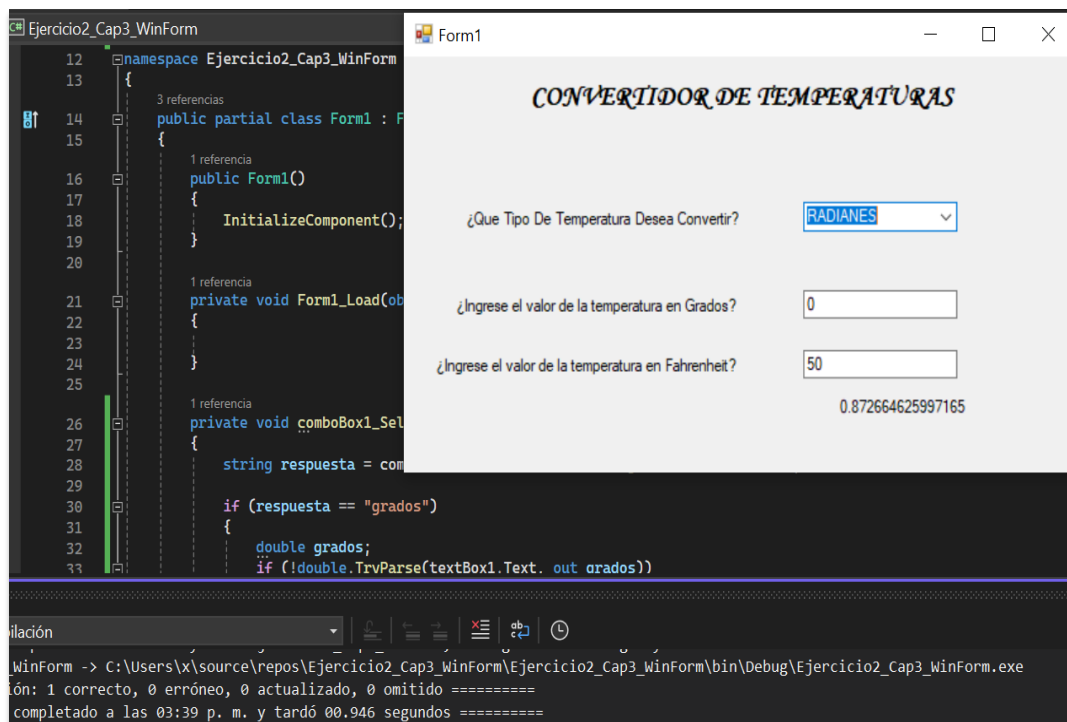
Ejercicio 2: Conversión entre grados y radianes

1. Interfaz de usuario:

- Dos cuadros de texto para que el usuario ingrese los grados y los radianes.
- Un botón para realizar la conversión.
- Dos etiquetas para mostrar los resultados de la conversión.

2. Funcionalidad:

- Cuando el usuario hace clic en el botón, se convierten los grados a radianes o viceversa.
- Los resultados se muestran en las etiquetas correspondientes.



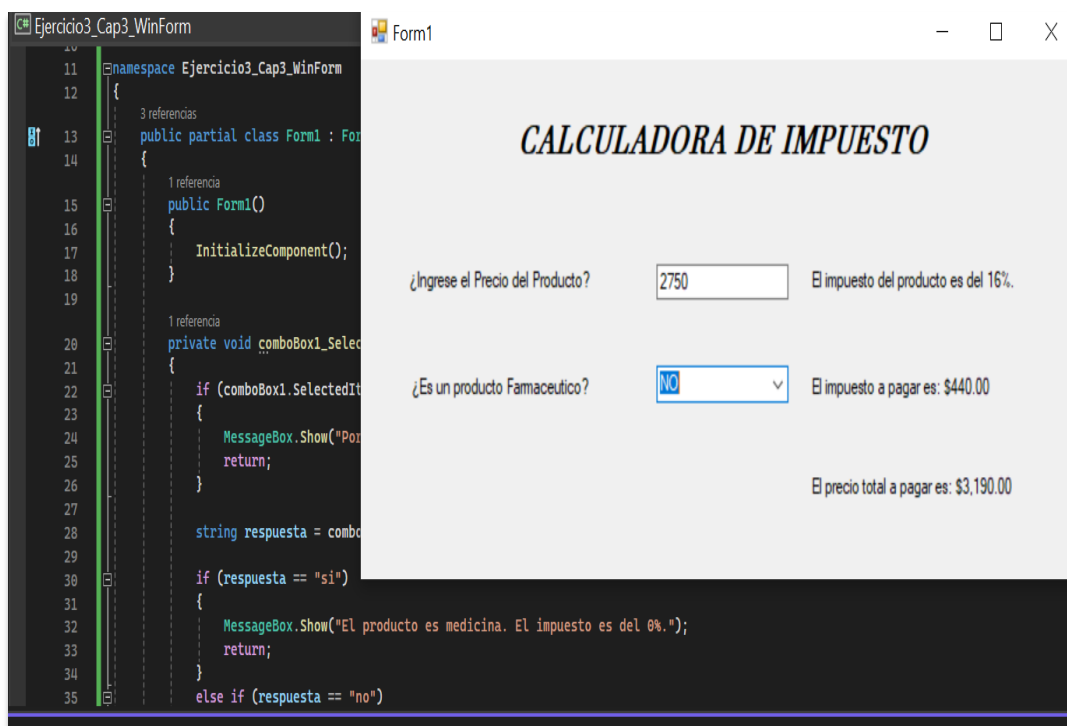
Ejercicio 3: Cálculo de impuestos para productos

1. Interfaz de usuario:

- Un cuadro de texto para que el usuario ingrese el precio del producto.
- Un cuadro de texto para que el usuario ingrese el tipo de producto (medicina o no).
- Un botón para calcular el impuesto.
- Etiquetas para mostrar el resultado del impuesto y el precio total.

2. Funcionalidad:

- Cuando el usuario hace clic en el botón, se calcula el impuesto según el tipo de producto.
- Los resultados se muestran en las etiquetas correspondientes.



Ejercicio 4: Asignar nombre a un día de la semana según un número

1. Interfaz de usuario:

- Un cuadro de lista desplegable para que el usuario seleccione un número del 1 al 7.
- Una etiqueta para mostrar el día de la semana correspondiente al número seleccionado.

2. Funcionalidad:

- Cuando el usuario selecciona un número, se determina el día de la semana correspondiente.
- El resultado se muestra en la etiqueta.



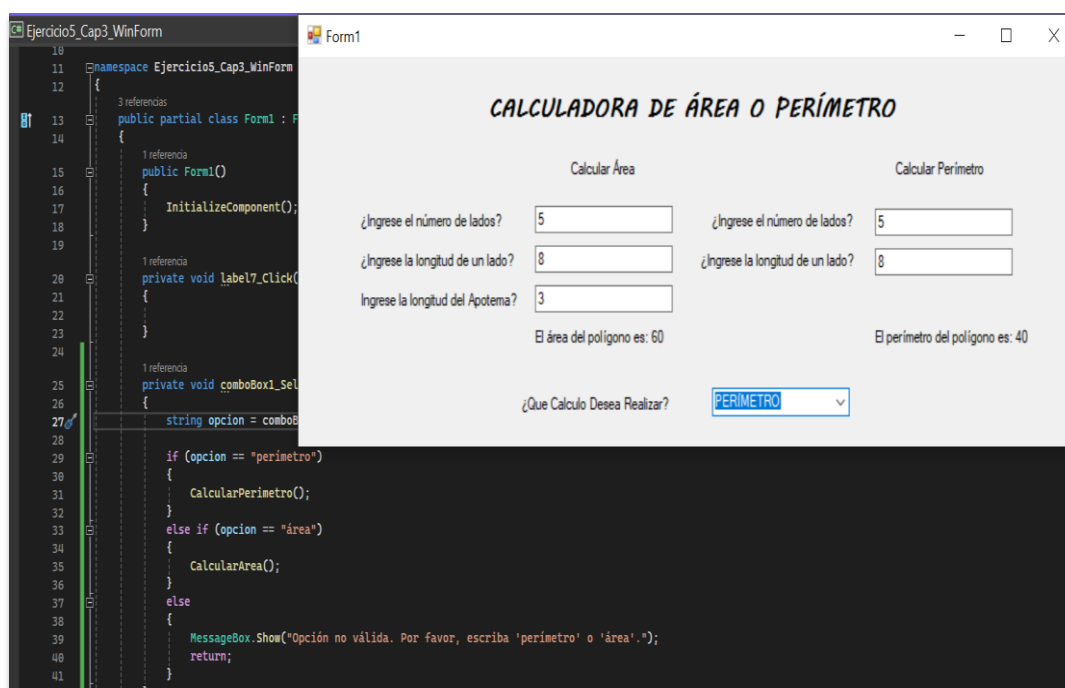
Ejercicio 5: Cálculo del perímetro y área de un polígono regular

1. Interfaz de usuario:

- Cuadros de texto para que el usuario ingrese el número de lados, la longitud del lado y la apotema del polígono.
- Un cuadro de lista desplegable para que el usuario seleccione si desea calcular el perímetro o el área.
- Un botón para realizar el cálculo.
- Etiquetas para mostrar los resultados del perímetro o el área.

2. Funcionalidad:

- Cuando el usuario hace clic en el botón, se calcula el perímetro o el área del polígono según la opción seleccionada.
- Los resultados se muestran en las etiquetas correspondientes.



6. RESULTADOS Y CONCLUSIONES

6.1 Resultados

Durante el desarrollo de esta práctica, se obtuvieron los siguientes resultados significativos:

Ejercicio 1: Determinar si un número es par o impar

- La aplicación permite al usuario ingresar un número.
- Al hacer clic en el botón de verificación, se determina si el número ingresado es par o impar.
- El resultado se muestra en la interfaz de usuario.

Ejercicio 2: Conversión entre grados y radianes

- La aplicación permite al usuario ingresar grados o radianes.
- Al hacer clic en el botón de conversión, se realiza la conversión y se muestran los resultados en la interfaz de usuario.

Ejercicio 3: Cálculo de impuestos para productos

- El usuario puede ingresar el precio del producto y especificar si es medicina o no.
- Al calcular el impuesto, se determina si es del 0% o del 16% según el tipo de producto.
- Los resultados del impuesto y el precio total se muestran en la interfaz de usuario.

Ejercicio 4: Asignar nombre a un día de la semana según un número

- El usuario puede seleccionar un número del 1 al 7.
- La aplicación determina el día de la semana correspondiente al número seleccionado.
- El resultado se muestra en la interfaz de usuario.

Ejercicio 5: Cálculo del perímetro y área de un polígono regular

- El usuario puede ingresar el número de lados, la longitud del lado y la apotema del polígono.
- Se puede seleccionar si se desea calcular el perímetro o el área.
- Al hacer clic en el botón de cálculo, se realiza el cálculo y se muestran los resultados en la interfaz de usuario.

6.2 Conclusiones

En conclusión, El desarrollo de aplicaciones con Windows Forms en C# permite crear interfaces de usuario intuitivas y funcionales.

Los ejercicios cubren diferentes aspectos de la programación, como la lógica de negocios, la manipulación de datos y el procesamiento de entrada del usuario.

La implementación de cada ejercicio proporciona una oportunidad para practicar y comprender conceptos clave de programación, como estructuras de control, manipulación de datos y manejo de eventos.

Los errores y desafíos encontrados durante el desarrollo ayudan a mejorar las habilidades de resolución de problemas y depuración.

La retroalimentación del usuario y la experiencia en la creación de interfaces de usuario pueden guiar futuras mejoras y optimizaciones en el diseño y la funcionalidad de las aplicaciones.

7. ANEXOS

Sin anexos.



8. REFERENCIAS