

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE TLAXIACO

DISEÑO DE ESTRUCTURA DE LIBRERÍA DLL

Unidad 2. Reporte de practica 2

**ASIGNATURA:
TOPICOS AVANZADOS DE PROGRAMACION**

**PRESENTA:
Edgar Axel Sandoval Hernández
Emmanuel Marx Salazar González**

**ASESOR
JOSE ALFREDO ROMAN CRUZ**

Tlaxiaco, Oax., 22 de marzo de 2024



"Educación, ciencia y tecnología, progreso día con día"®

ÍNDICE

| | |
|---|----|
| OBJETIVO GENERAL | 2 |
| MATERIALES..... | 2 |
| INTRODUCCIÓN | 3 |
| MARCO TEÓRICO | 3 |
| ¿Qué es una DLL? | 3 |
| La estructura de una DLL en C# consta de varios elementos importantes, como se detalla a continuación: | 3 |
| RESULTADOS..... | 14 |
| CONCLUSIÓN | 15 |
| BIBLIOGRAFÍAS..... | 15 |

OBJETIVO GENERAL

El objetivo de esta práctica es realizar la consulta de la estructura de una librería en nuestro caso (DLL) e implementar funciones con las librerías en una App de escritorio (calculadora), utilizando las herramientas del IDE Visual Studio y dar a conocer los pasos a seguir para poder llegar a lo previsto.

MATERIALES

- Equipo de cómputo (Laptop)
- Internet
- Luz
- IDE de Visual Studio

INTRODUCCIÓN

En esta práctica se realizó la implementación de una librería para ejecutarlo desde consola, lo cual esto nos ayudó para tener más conocimiento acerca de la creación de las librerías en visual Studio. Para poder desarrollar nuestro proyecto en consola se creó la librería en una biblioteca de clase, posteriormente se agregaron los códigos para mandar a llamar los métodos y agregarle las referencias al proyecto y después compilar y ejecutarlo para saber que no tenga algún error.

Una creación de una librería nos proporciona funcionalidades comunes, que ya hemos resuelto previamente y así se evita la duplicidad del código al igual se utiliza para desarrollar software, desde la primera versión Visual Studio nos ha permitido hacer modificaciones mediante el uso de Git Bash para agregar o modificar el código que nos genera en Visual y así en equipo trabajar equitativamente y no perder el trabajo realizado.

MARCO TEÓRICO

¿Qué es una DLL?

Una DLL (Dynamic Link Library) en C# es un archivo que contiene un código compilado que se puede utilizar en varias aplicaciones.

Una DLL se utiliza específicamente para proporcionar funcionalidades reutilizables en diferentes aplicaciones, lo que la hace una herramienta importante para la reutilización de código y la modularidad del diseño del software.

La estructura de una DLL en C# consta de varios elementos importantes, como se detalla a continuación:

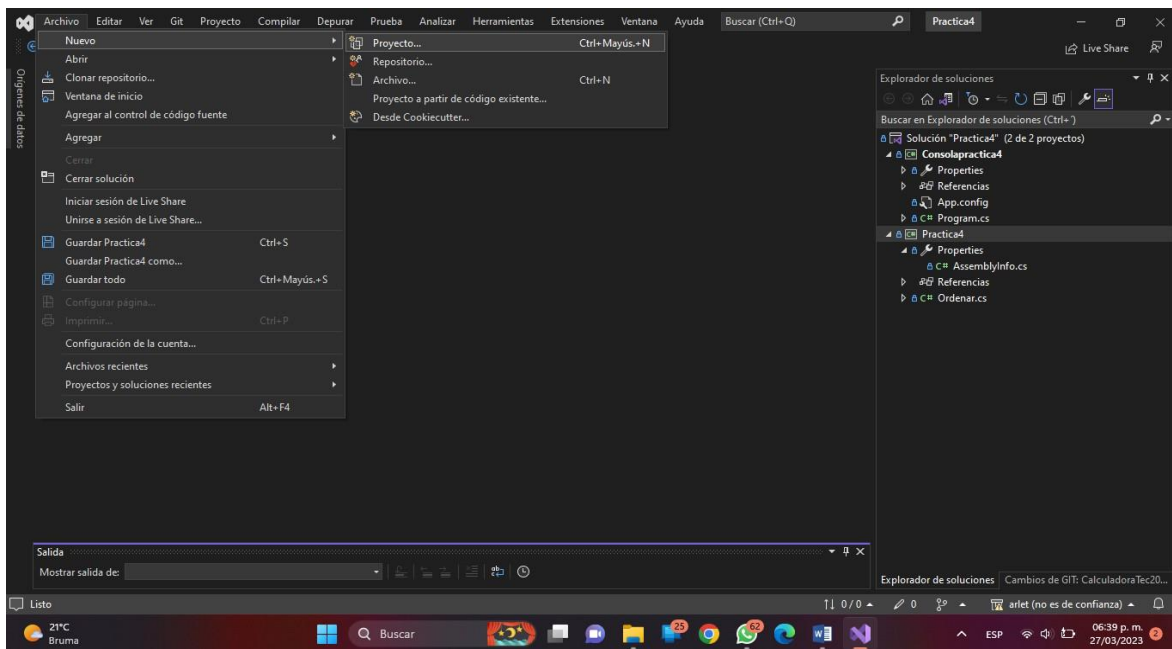
1. **Encabezado de la DLL:** Es la primera sección del archivo DLL y contiene información sobre el archivo DLL, como la versión, el tipo de archivo y el tamaño.
2. **Tabla de exportación:** Esta sección contiene información sobre las funciones y los datos que se pueden utilizar desde fuera de la DLL. La tabla de exportación también contiene los nombres de las funciones que se pueden llamar desde otras aplicaciones.
3. **Tabla de importación:** Esta sección contiene información sobre las funciones y los datos que se deben importar desde otras bibliotecas para que

la DLL pueda funcionar correctamente. La tabla de importación también contiene los nombres de las funciones y las bibliotecas que se deben importar.

4. **Código compilado:** Esta sección contiene el código compilado que se puede utilizar en varias aplicaciones. El código compilado incluye todas las funciones y los datos definidos en la DLL.
5. **Sección de recursos:** Esta sección contiene información adicional que se puede utilizar en la aplicación, como iconos, gráficos, archivos de sonido y otros tipos de datos.

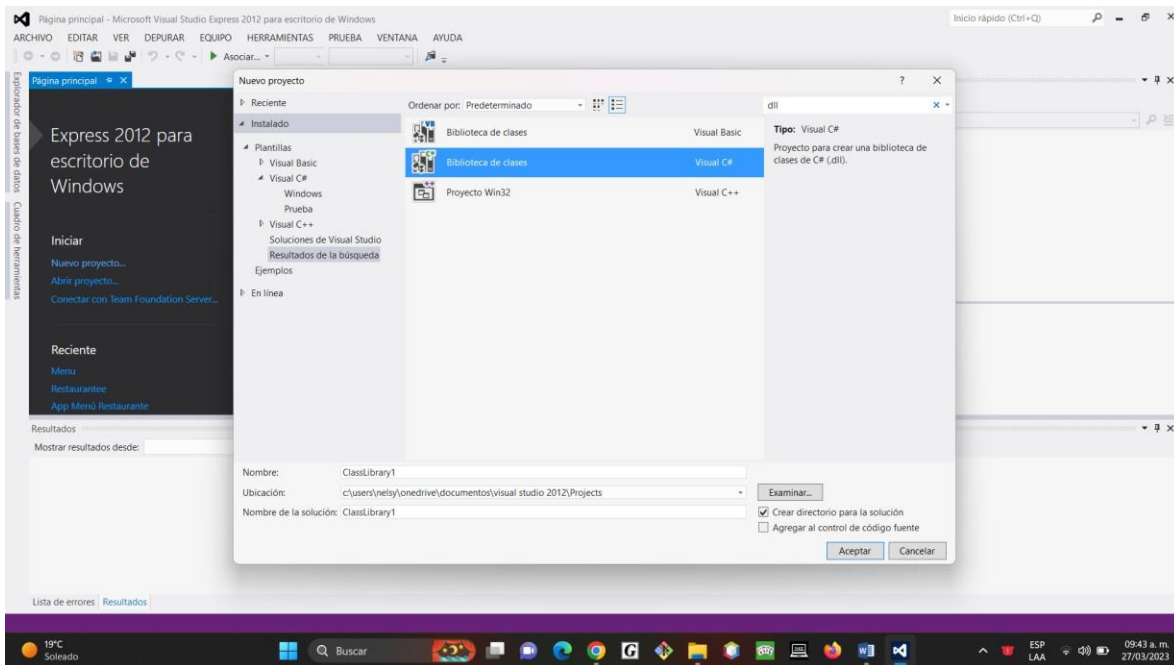
MANUAL PARA CREAR UNA APLICACIÓN DE CONSOLA (Tomar la orden de un cliente en un restaurante)

1. Primeramente creamos un **nuevo proyecto** en Visual Studio, nos redirigimos en el comando de **Archivo** y le damos click, en seguida en **Nuevo** y finalmente en **Proyecto**.

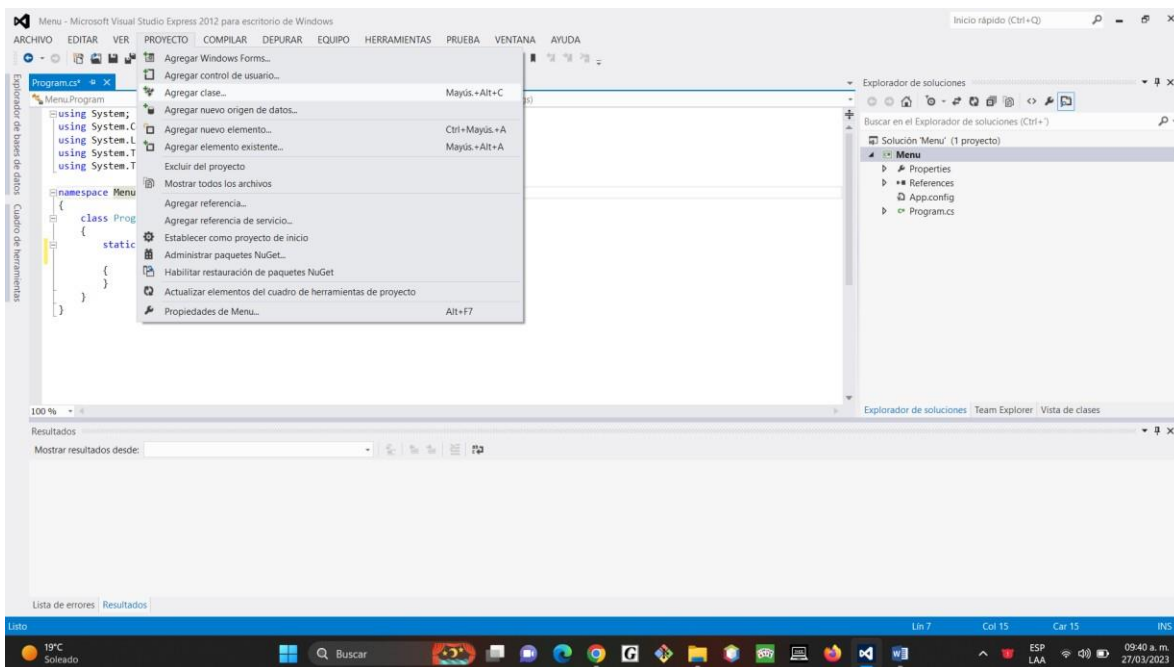


2. En seguida nos aparece el siguiente cuadro de dialogo, buscamos la opción de **Biblioteca de Clases con .dll**, y le damos click, configuramos el **nombre**

de nuestro proyecto dependiendo nuestras preferencias, una vez realizado lo anteriormente descrito le damos click en **Aceptar**.



- Posteriormente nos redirigimos en el comando de **Proyecto** como se muestra en la imagen y le damos click, en seguida buscamos la opción de agregar clase.



4. A continuación, le asignamos un **Nombre** a nuestra clase creada dependiendo a nuestras necesidades.

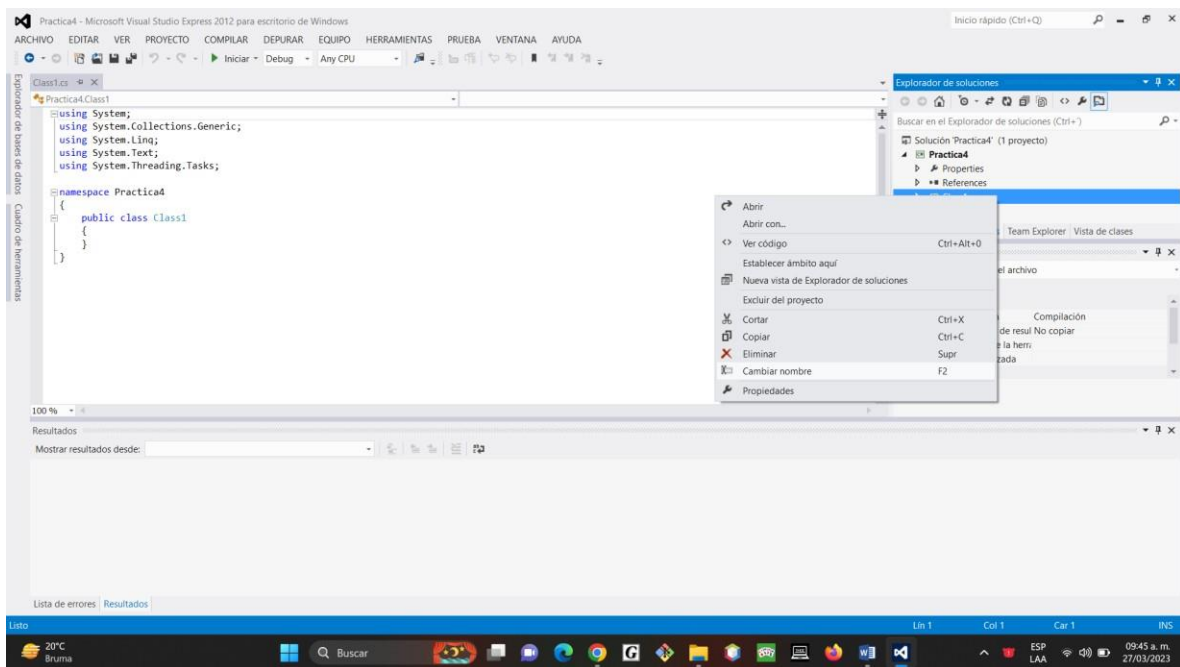


Ilustración 1

En nuestro caso le asignamos el nombre de Practica4.

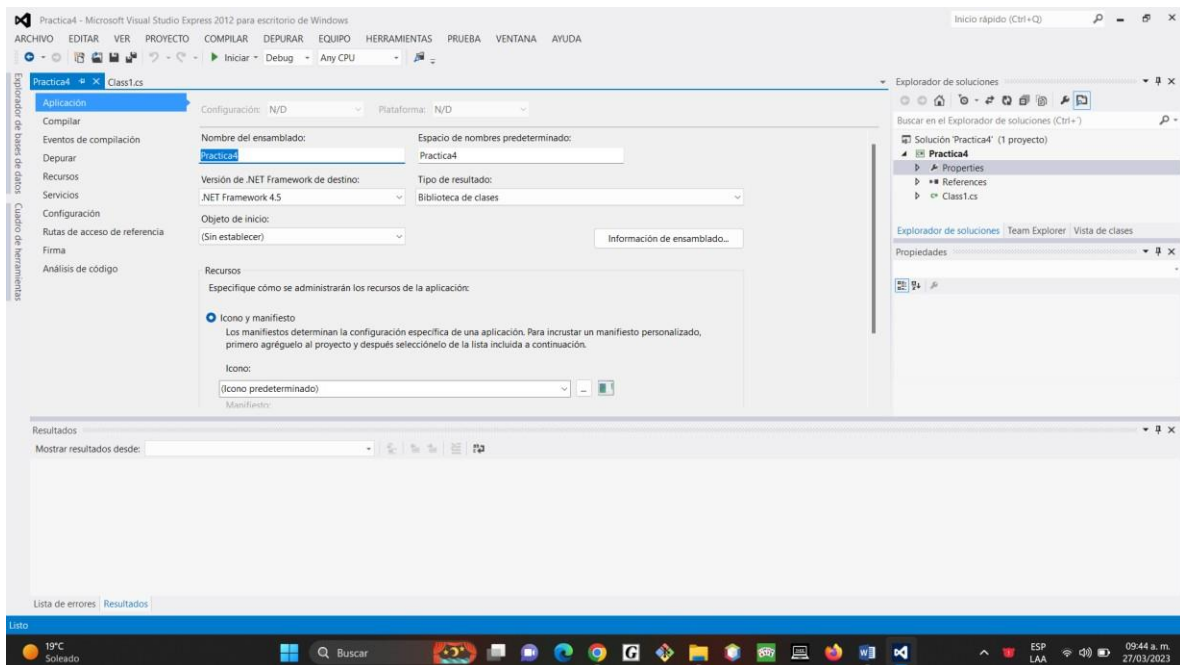


Ilustración 2

5. Después nos aparece el siguiente cuadro de dialogo, para confirmar le damos click en la **opción Si**.

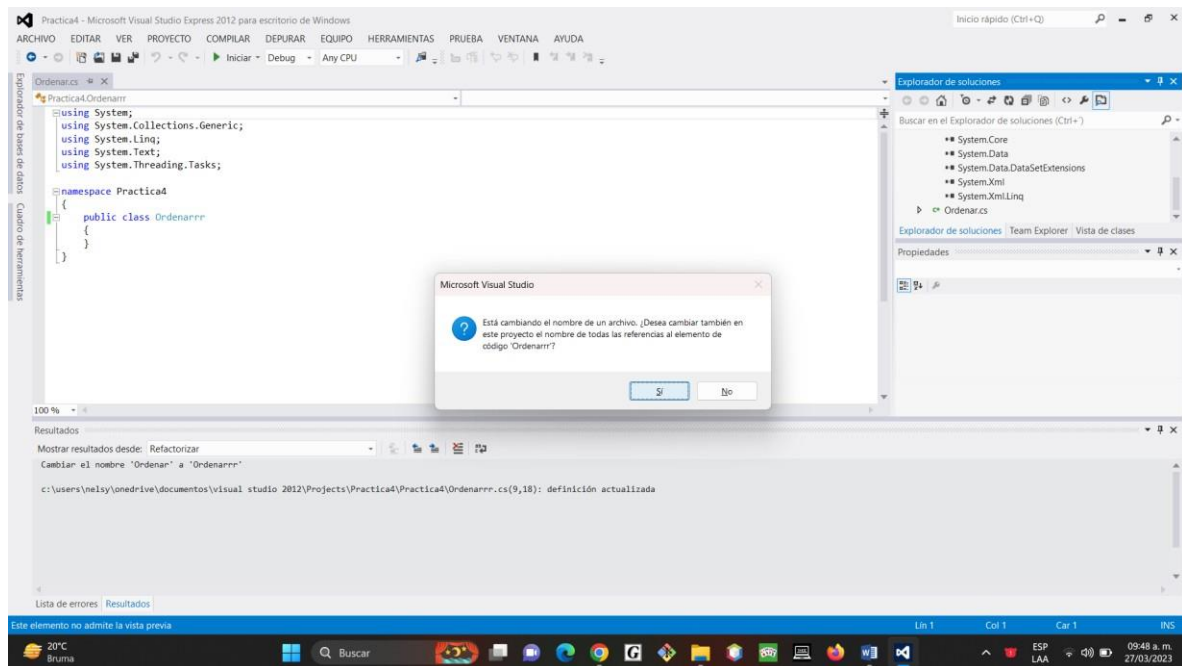


Ilustración 3

6. Una vez creada la clase iniciamos a **codificar dependiendo a nuestro proyecto**, en nuestro caso para ordenar en un restaurante, mostrando nuestro código en la siguiente imagen.

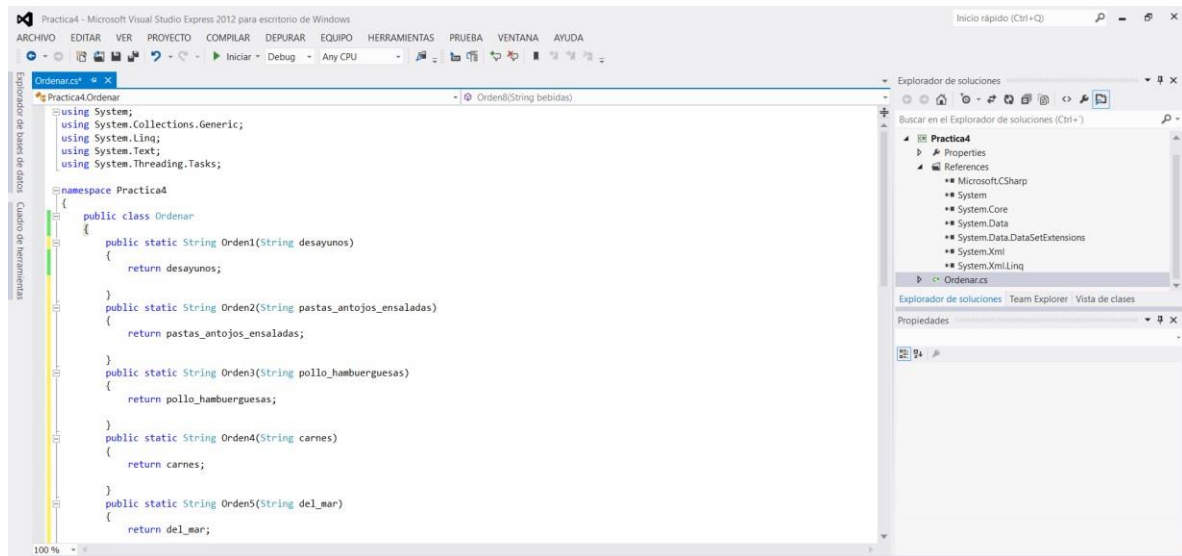


Ilustración 4

7. Posteriormente una vez culminado el código lo **ejecutamos** para asegurarnos que no contenga ningún error.

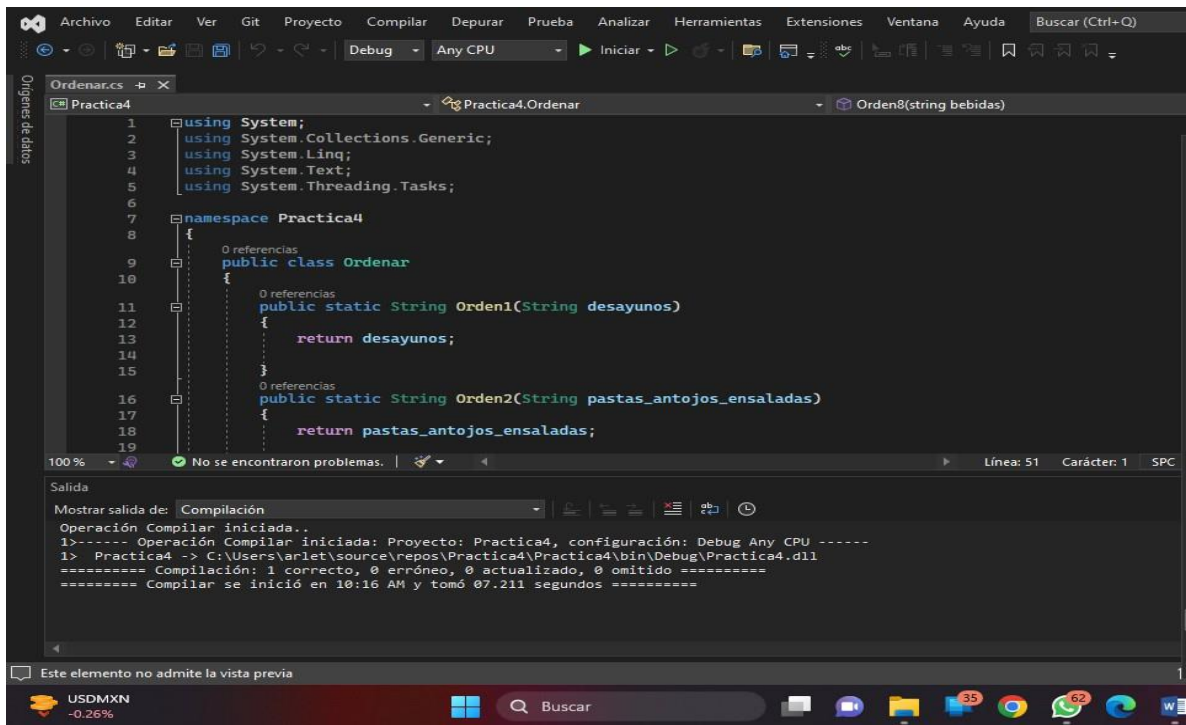


Ilustración 5

8. En seguida ya que hayamos comprobado de que si compila nuestro código, le damos **click derecho** sobre la opción de **properties** y nos arrojará el siguiente cuadro de dialogo, buscamos la opción de **abrir carpeta en el explorador de Archivos** y le damos click.

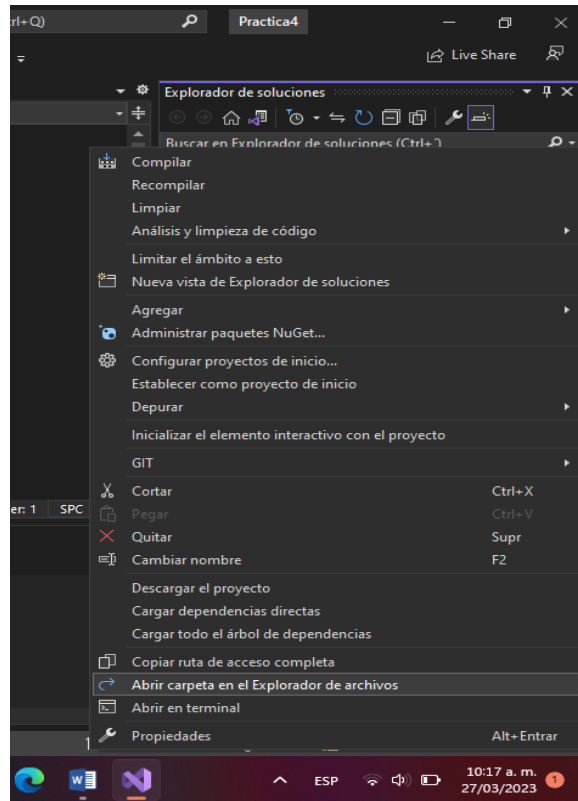


Ilustración 6

9. Una vez realizado lo anterior el IDE en automático nos redirige a la carpeta dependiendo donde se haya almacenado nuestra dll, finalmente la seleccionamos.

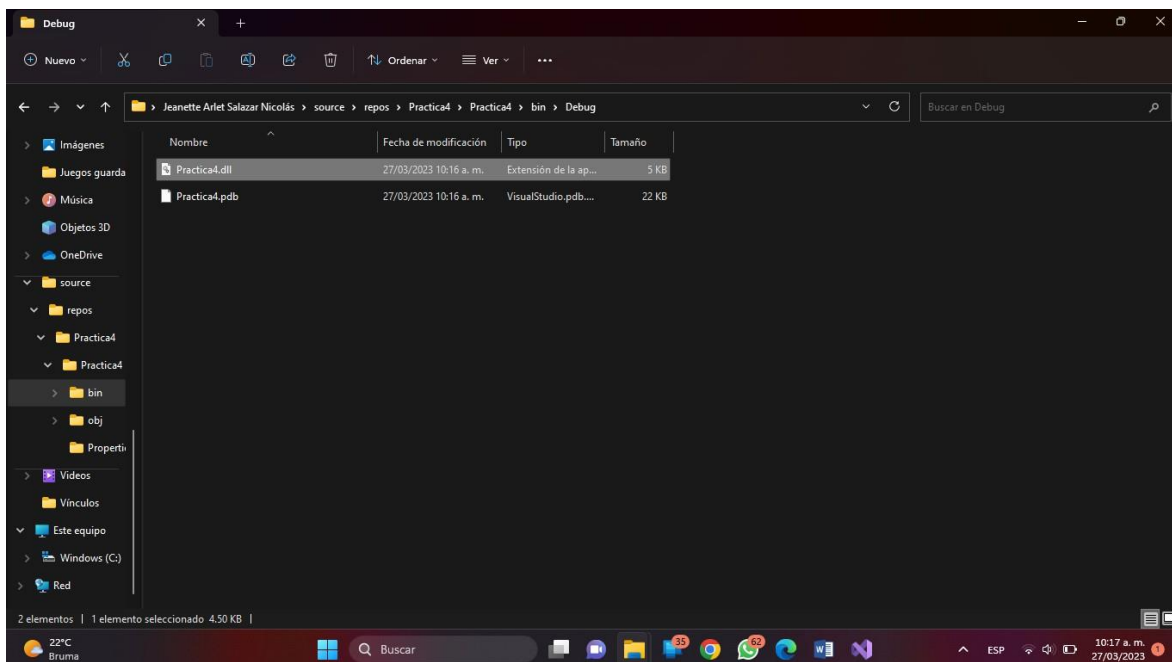


Ilustración 7

10. Finalmente ya que se haya realizado todo lo anteriormente descrito, le damos nuevamente click derecho sobre la opción de properties y buscamos la opción de agregar, esto con la finalidad de agregar un nuevo proyecto.

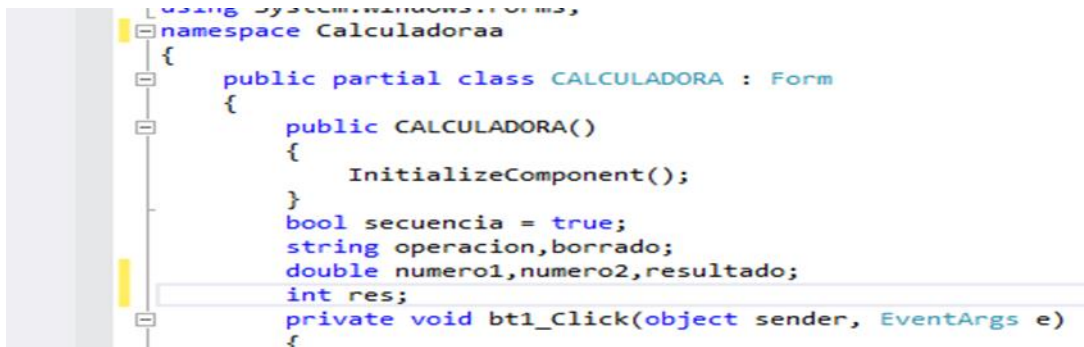


Ilustración 8

11. A continuación reelegimos la opción de Aplicación de consola siempre y cuando revisemos que sea .Net y dll.

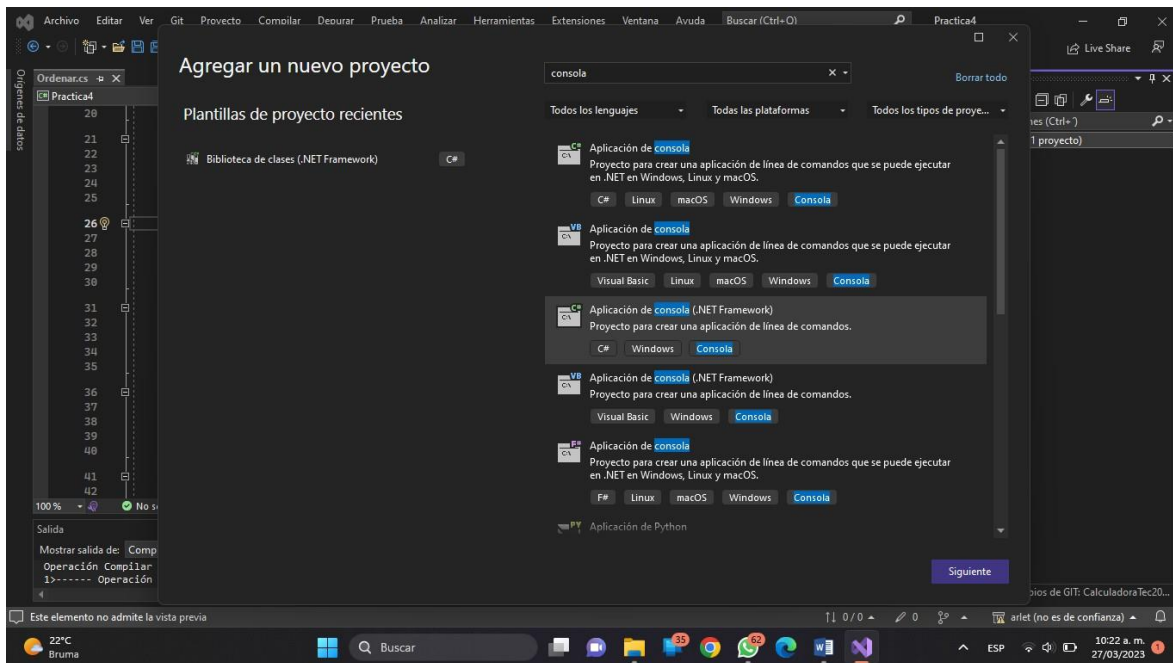


Ilustración 9

12. Le asignamos un nombre a nuestro nuevo proyecto.

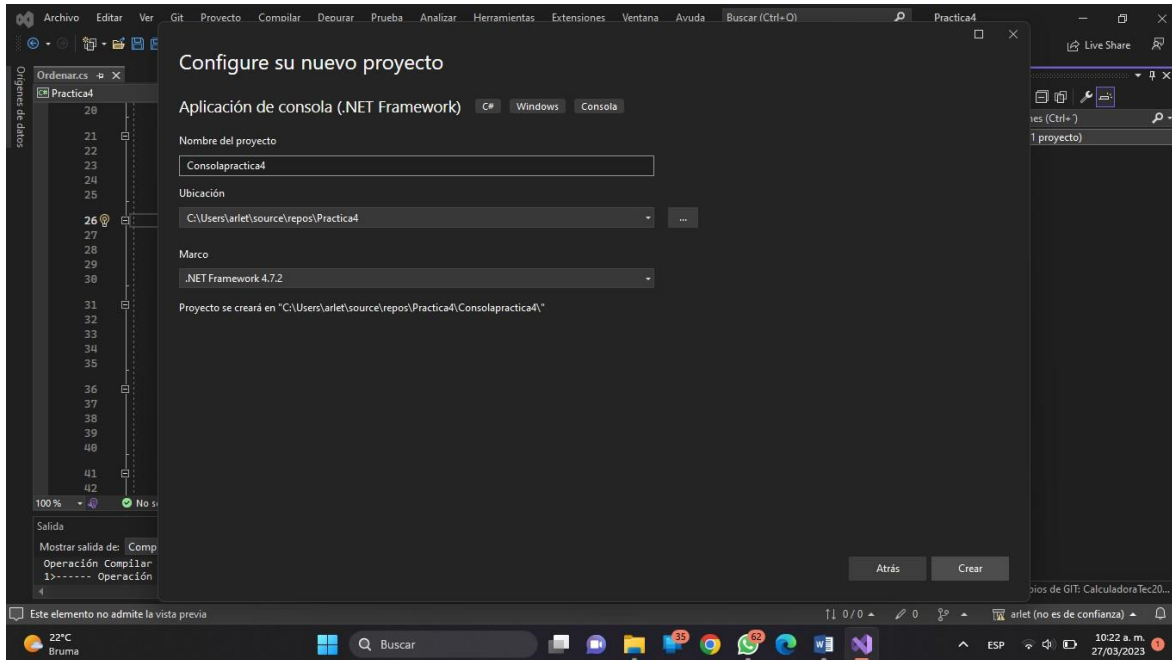


Ilustración 10

13. Después le damos nuevamente click derecho sobre la opción de properties y buscamos la opción de Establecer como proyecto de inicio como se muestra en la imagen.

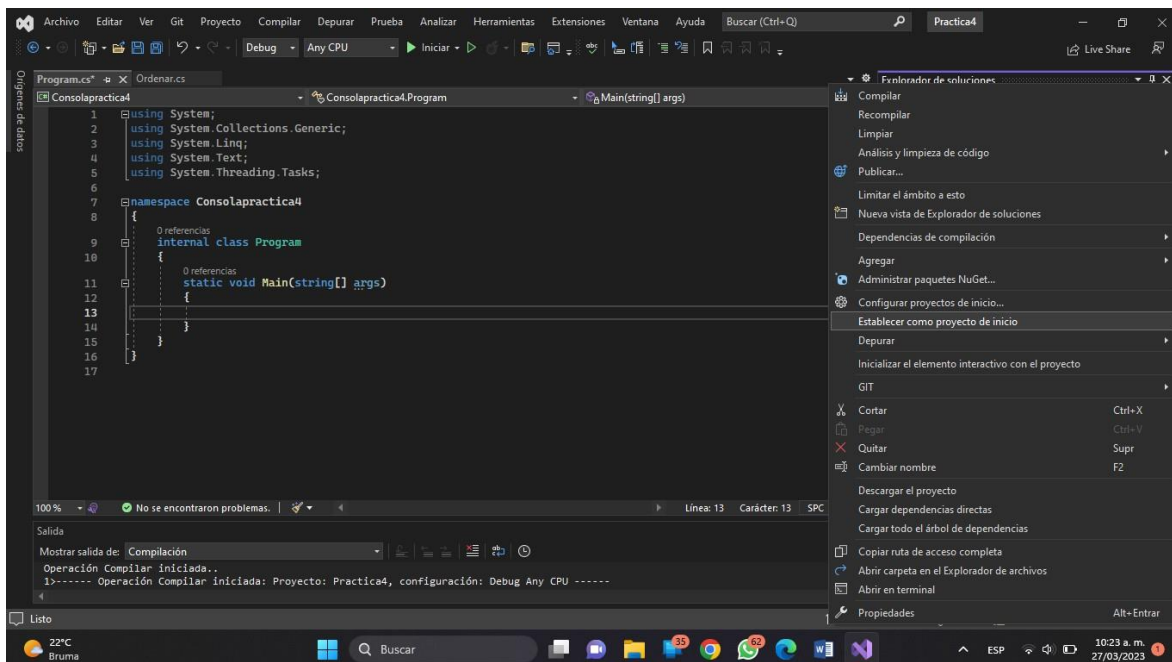


Ilustración 11

14. En seguida, le damos click derecho sobre la opción de referencias y buscamos la opción de Agregar referencia y le damos click.

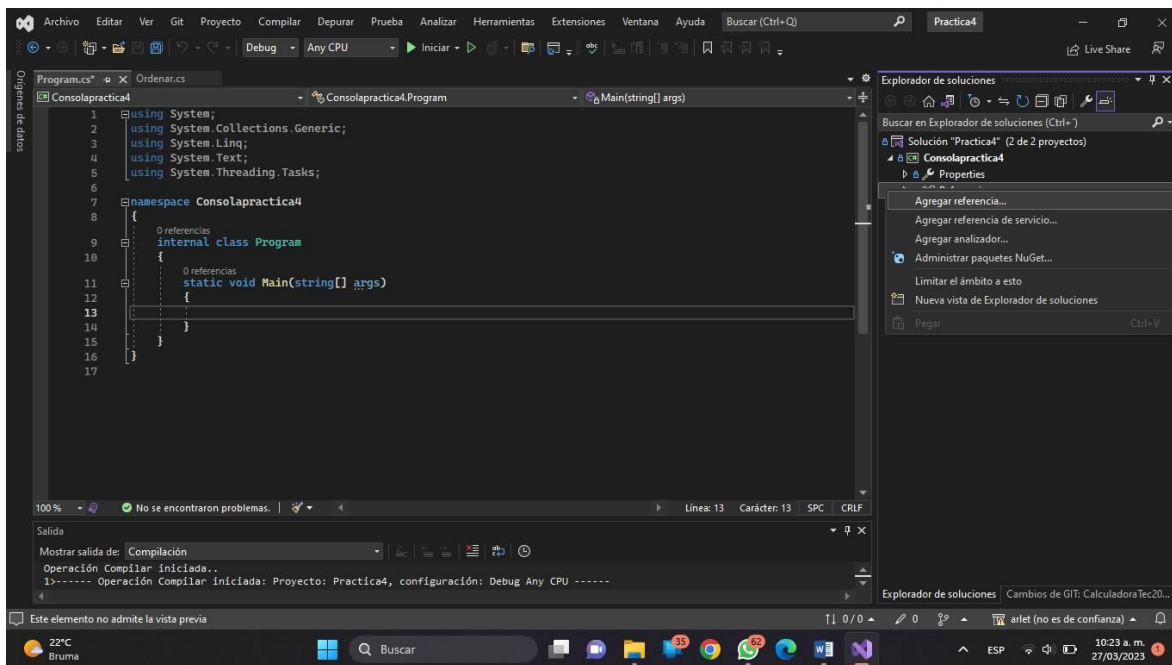


Ilustración 12

15. Posteriormente nos arroja el siguiente cuadro de dialogo, nos dirigimos a la opcion de examinar y le damos click.

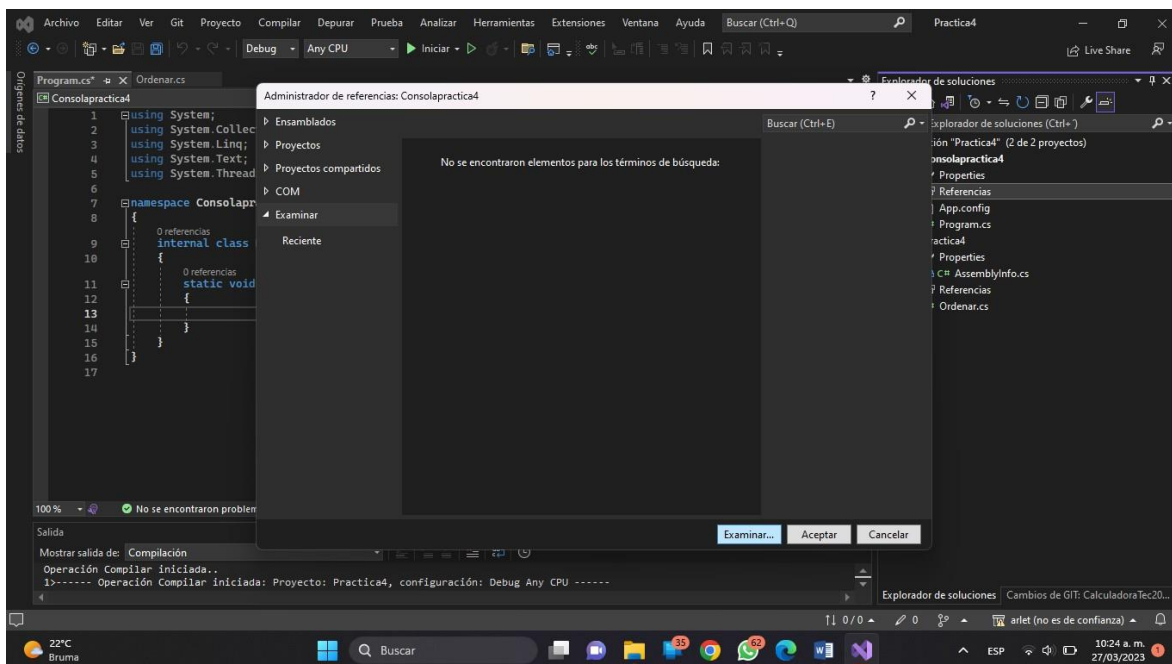


Ilustración 13

16. A continuación el IDE nos redirige a la siguiente ventana, en donde elegimos la opción que nos sugiere, una vez realizado eso le damos click en aceptar.

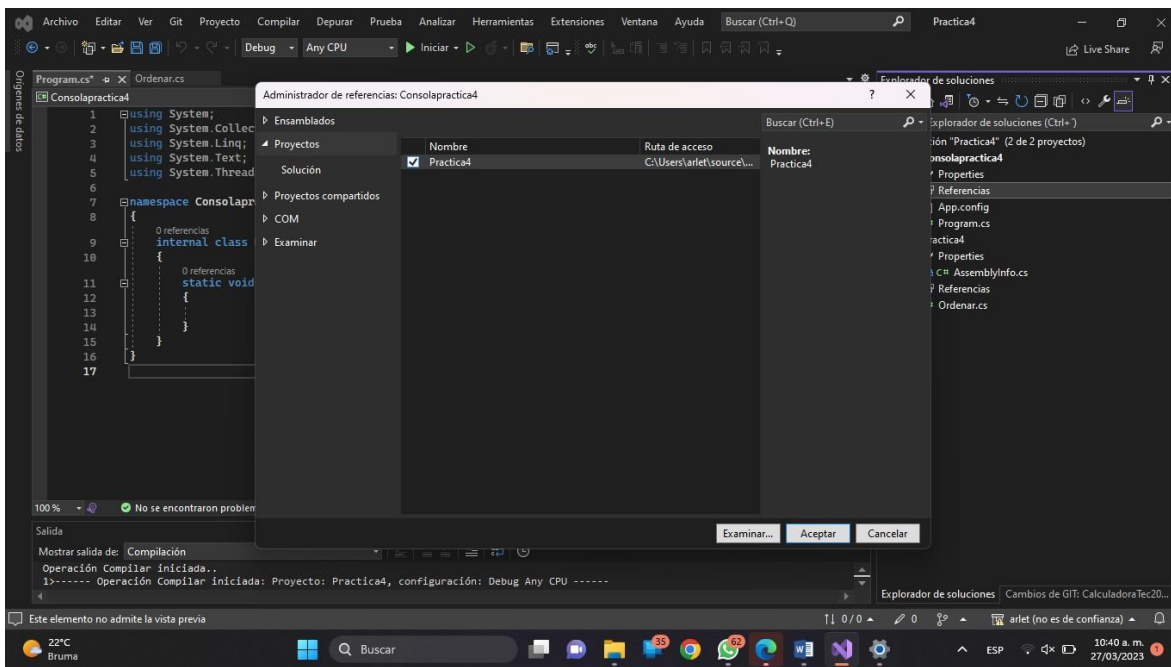


Ilustración 14

17. Una vez culminado los pasos anteriores iniciamos a codificar nuevamente, en nuestro caso nos queda de la siguiente manera.

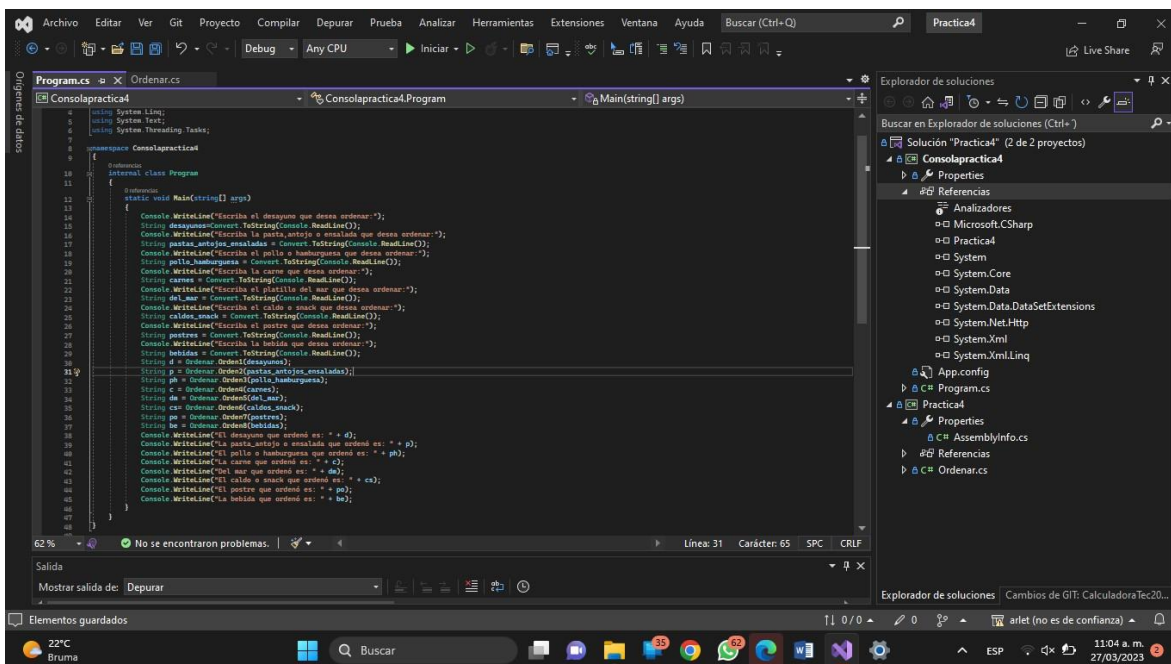


Ilustración 15

RESULTADOS

18.-Por último, una vez culminado nuestro código lo ejecutamos y rellenamos los parámetros que se solicitan.

```
public double Coseno (double a)
{
    return Math.Cos(a);
}

private void Resultado_TextChanged(object sender, EventArgs e)
{
}

private void CALCULADORA_Load(object sender, EventArgs e)
{
}

private void btponeto_Click(object sender, EventArgs e)
{
}

}
```

Ilustración 16

19.-Finalmente nuestra aplicación está concluida.

| | |
|---------------------|--------------------------------------|
| Ilustración 1..... | 6 |
| Ilustración 2..... | 6 |
| Ilustración 3..... | 7 |
| Ilustración 4..... | 7 |
| Ilustración 5..... | 8 |
| Ilustración 6..... | 9 |
| Ilustración 7..... | 9 |
| Ilustración 8..... | 10 |
| Ilustración 9..... | 10 |
| Ilustración 10..... | 11 |
| Ilustración 11..... | 11 |
| Ilustración 12..... | 12 |
| Ilustración 13..... | 12 |
| Ilustración 14..... | 13 |
| Ilustración 15..... | 13 |
| Ilustración 16..... | ¡Error! Marcador no definido. |

CONCLUSIÓN

Crear una DLL en una aplicación de consola en C# es un proceso sencillo. Solo necesitamos agregar una clase pública al proyecto de consola, compilarlo y agregar una referencia a la DLL en cualquier proyecto en el que se desee utilizar.

La diferencia principal entre una DLL y una aplicación de consola es que una DLL es un archivo de biblioteca de enlace dinámico que se puede utilizar en diferentes proyectos, mientras que una aplicación de consola es un programa independiente que se ejecuta en una consola de comandos.

Aprendimos a crear bibliotecas de clases, exponer clases y métodos públicos, agregar referencias de ensamblado, compilar y distribuir DLL, y depurar el código fuente.

La DLL es una forma de encapsular el código y reutilizarlo en diferentes proyectos, lo que puede ahorrar tiempo y esfuerzo.

Para trabajar con DLL en C#, es importante comprender su función, cómo se utilizan en el desarrollo de software, cómo exponer clases y métodos públicos, cómo agregar referencias de ensamblado, cómo depurar y cómo optimizar el rendimiento de la aplicación al utilizar DLL.

BIBLIOGRAFÍAS

<http://luisvera.com/net-c-tutorial-como-crear-una-dll-y-como-utilizarla/#:~:text=Una%20biblioteca%20de%20enlace%20din%C3%A1mico,los%20sistemas%20operativos%20Windows%20siendo%20%E2%80%9C>