

INGENIERÍA MECATRÓNICA



DI\_CERO

DIEGO CERVANTES RODRÍGUEZ

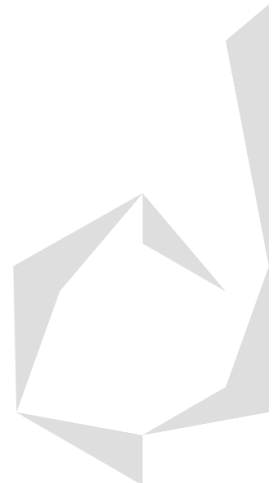
INSTRUMENTACIÓN VIRTUAL

C# Y VISUAL STUDIO 2019

Introducción al Lenguaje de  
Programación C#

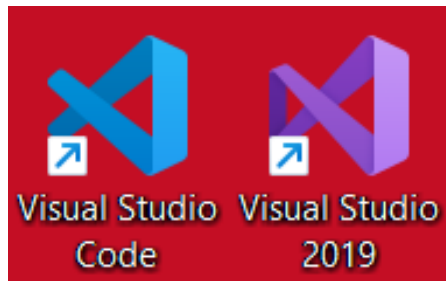
# Contenido

C# - <b>Visual Studio</b> (Logo Morado).....	2
Aplicación de consola (.NET Framework): .....	4
Aplicación de Windows Forms (.NET Framework):.....	5
Agregar Librerías a C# en Visual Studio: .....	6
Creación de GUI en Visual Studio <b>Aplicación de Windows Forms (.NET Framework)</b> : .....	7

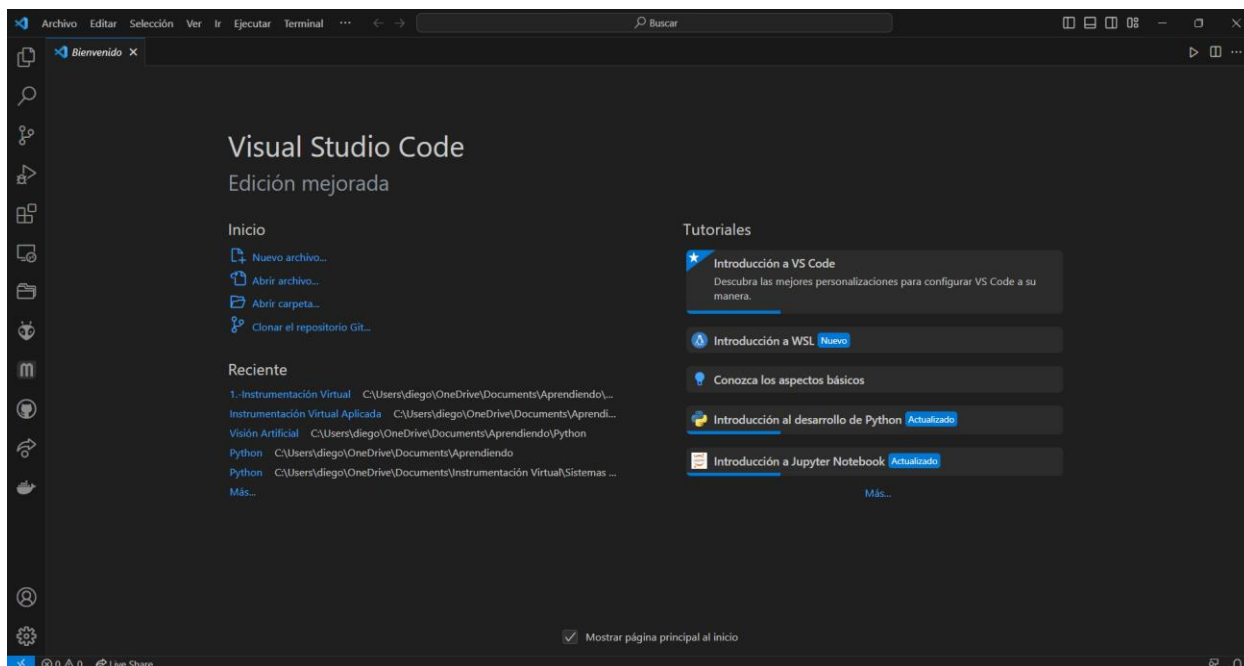


# C# - Visual Studio (Logo Morado)

Para ejecutar los programas creados con el lenguaje de programación C# se debe utilizar el editor de texto Visual Studio, que no es igual a Visual Studio Code, ya que llega a haber confusión entre estos dos.



- Su principal diferencia radica en que **Visual Studio Code** es un **editor** muy versátil que ejecuta **texto plano**, pero para ello se deben importar Extensiones que permitan al editor interpretar los distintos lenguajes de programación que se le pida.





- Mientras que **Visual Studio** está mas enfocado en crear **proyectos robustos de código**, permitiendo así que se puedan gestionar dentro del mismo editor de texto los elementos gráficos (widgets) incluidos en una interfaz gráfica (GUI), conexión con bases de datos, etc. Para ello se puede elegir una gran variedad de plantillas que permiten crear proyectos de todo tipo, pero los principales que se utilizan son los siguientes:
  - **Aplicación de consola (.NET Framework):** Es una plantilla de Visual Studio que permite crear programas de consola muy sencillos, donde su resultado solamente podrá ser visto en consola.
  - **Aplicación de Windows Forms (.NET Framework):** Es una plantilla de Visual Studio que permite crear interfaces gráficas con las que podrá interactuar el usuario, si un

programa se crea con esta plantilla, desde un inicio todas sus funciones se irán programando sobre los elementos gráficos y cada uno de ellos podrá ser agregado por medio de Visual Studio.

## Crear un proyecto

### Plantillas de proyecto recientes

 Aplicación de Windows Forms (.NET Framework) C#

 Aplicación de consola (.NET Framework) C#

Todos los lenguajes

Todas las plataformas

Todos los tipos de proy...



#### Aplicación de consola

Proyecto para crear una aplicación de línea de comandos que se puede ejecutar en .NET Core en Windows, Linux y macOS.

C#

Linux

macOS

Windows

Consola



#### Aplicación de consola

Proyecto para crear una aplicación de línea de comandos que se puede ejecutar en .NET Core en Windows, Linux y macOS.

Visual Basic

Linux

macOS

Windows

Consola



#### Aplicación web de ASP.NET Core

Una plantilla de proyecto para crear una aplicación ASP.NET Core con contenido de páginas de Razor de ASP.NET de ejemplo.

C#

Linux

macOS

Windows

Nube

Servicio

Web



#### Aplicación Blazor para WebAssembly

Plantilla de proyecto para crear una aplicación Blazor que se ejecuta en WebAssembly y que se hospeda de forma opcional en una aplicación de ASP.NET Core. Esta plantilla se puede usar para las aplicaciones web con interfaces de usuario dinámicas enriquecidas (IU).

C#

Linux

macOS

Windows

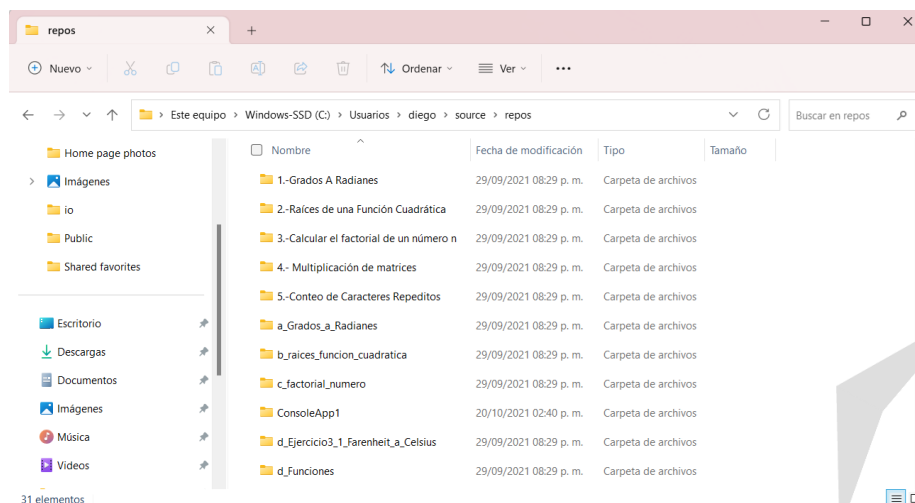
Nube

Web

Atrás

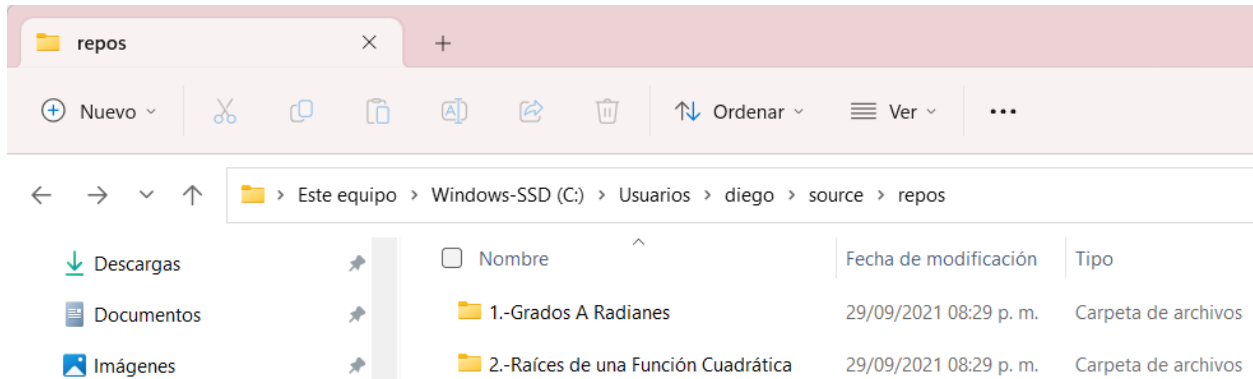
Siguiente

- Además, siempre que se creen programas con Visual Studio, se creará una carpeta del proyecto, sin la cual el programa no se podrá ejecutar, sea cual sea la plantilla que se haya elegido.
  - Todas las carpetas de los proyectos creados con Visual Studio se almacenarán por default en el directorio: **C:\Users\diego\source\repos**



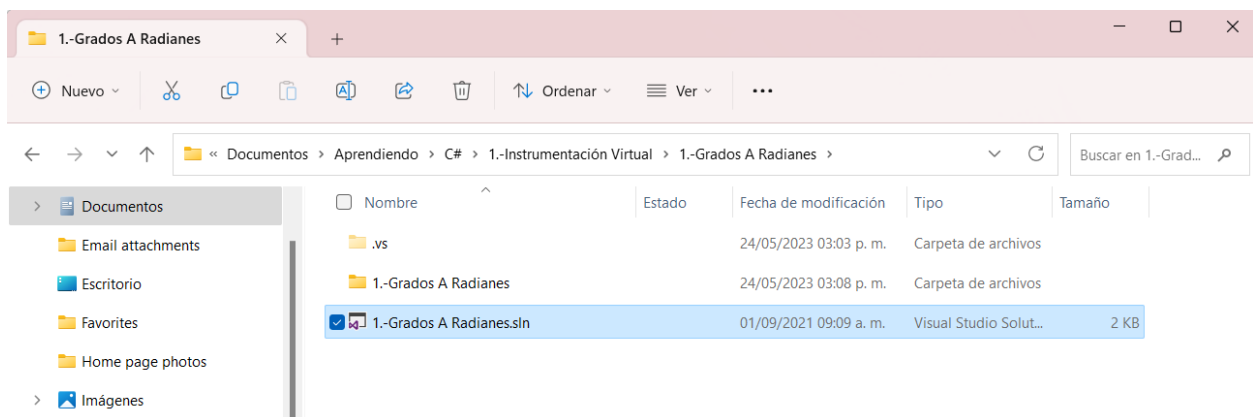
## Aplicación de consola (.NET Framework):

Cuando se generen programas con esta plantilla se creará una carpeta del proyecto, no se puede eliminar esta carpeta, aunque si se le puede cambiar el nombre solo de afuera, sin haber cambiado el del archivo con extensión .sln, el cual es el que abre Visual Studio para poder ejecutar el código.

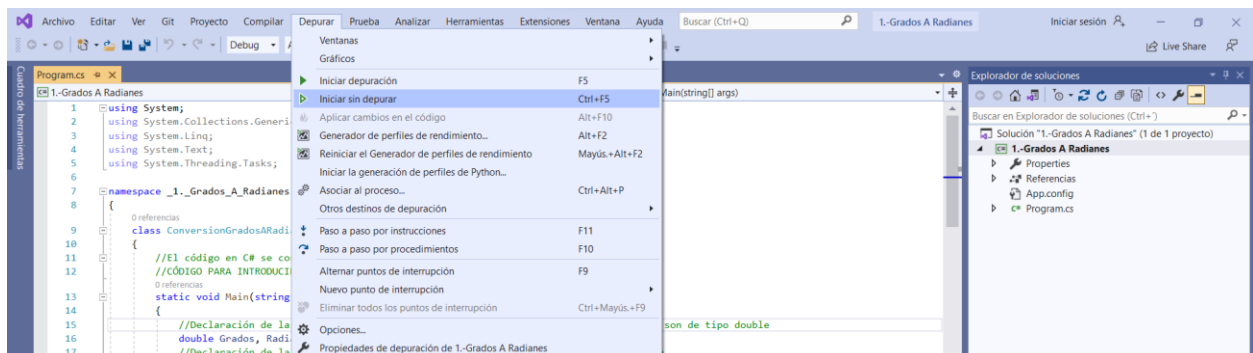


***Nota: Aunque por defecto los programas hechos en Visual Studio se crean dentro del directorio: C:\Users\diego\source\repos***

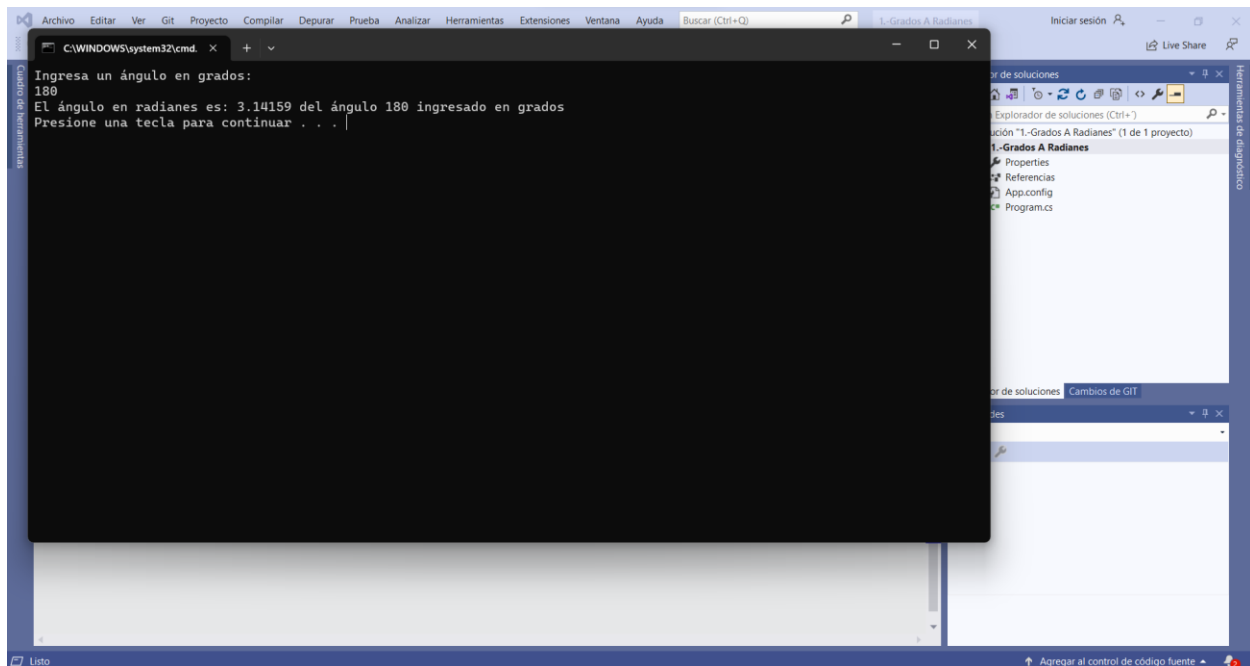
***No forzosamente se deben encontrar ahí para que puedan ser ejecutados.***



Una vez que se haya dado doble clic sobre el archivo con extensión .sln, se abrirá el editor de texto Visual Studio, para ejecutar un programa de consola dentro de él, se debe seleccionar la opción de: Depurar → Iniciar sin Depurar.



Al hacerlo se ejecutará correctamente el programa sin cerrar la consola, ya que si se da simplemente clic en el botón de Iniciar que tiene el símbolo de Play, la consola se cerrará una vez que el código interactúe con ella.



## Aplicación de Windows Forms (.NET Framework):

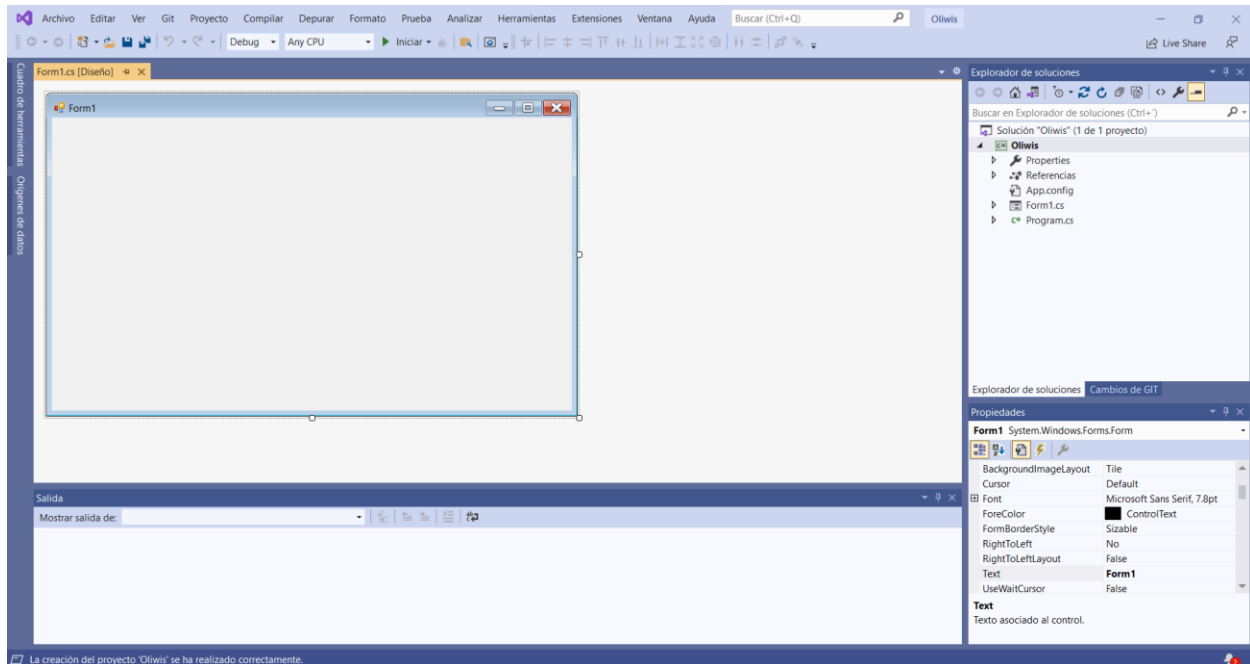
Esta plantilla sirve para crear programas que se busca presentar por medio de una GUI (Graphical User Interface).

Las interfaces gráficas o GUI por sus siglas en inglés son ventanas con elementos gráficos con los que puede interactuar el usuario como botones, áreas de texto, controles de texto, desplegables, listas, controles numéricos, imágenes, gráficas, etc. Esto sirve para realizar cualquier acción que se quiera ejecutar de forma gráfica con un código hecho enteramente con el lenguaje de programación C#, esto no se puede realizar con la forma simple de C# por lo que se debe de utilizar la plantilla **Aplicación de Windows Forms (.NET Framework)**, perteneciente al editor de texto Visual Studio.

La terminología de las GUI es:

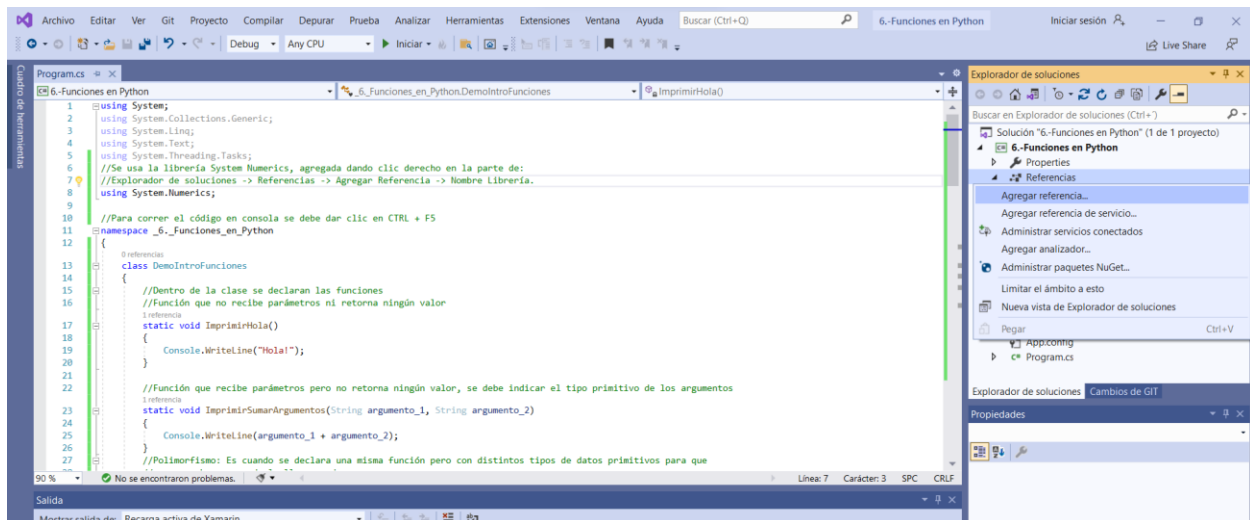
- **Método main:** Es un método en el lenguaje de programación C# a través del cual se ejecutan los métodos de todas las clases incluidas en el programa.
- **Frame:** Es una clase perteneciente al lenguaje de programación C# que sirve para crear la ventana de la GUI.
- **Panel:** Es una clase perteneciente al lenguaje de programación C# que sirve para crear los contenedores que se encuentran dentro del **Frame** de la GUI y que a su vez contienen los elementos gráficos con los que va a interactuar el usuario llamados **Widgets**.

- **Widgets:** Son los botones, áreas de texto, controles de texto, checkbox, radio buttons, desplegables, listas, controles numéricos, imágenes, gráficas, etc. con los que interactúa el usuario en la GUI.



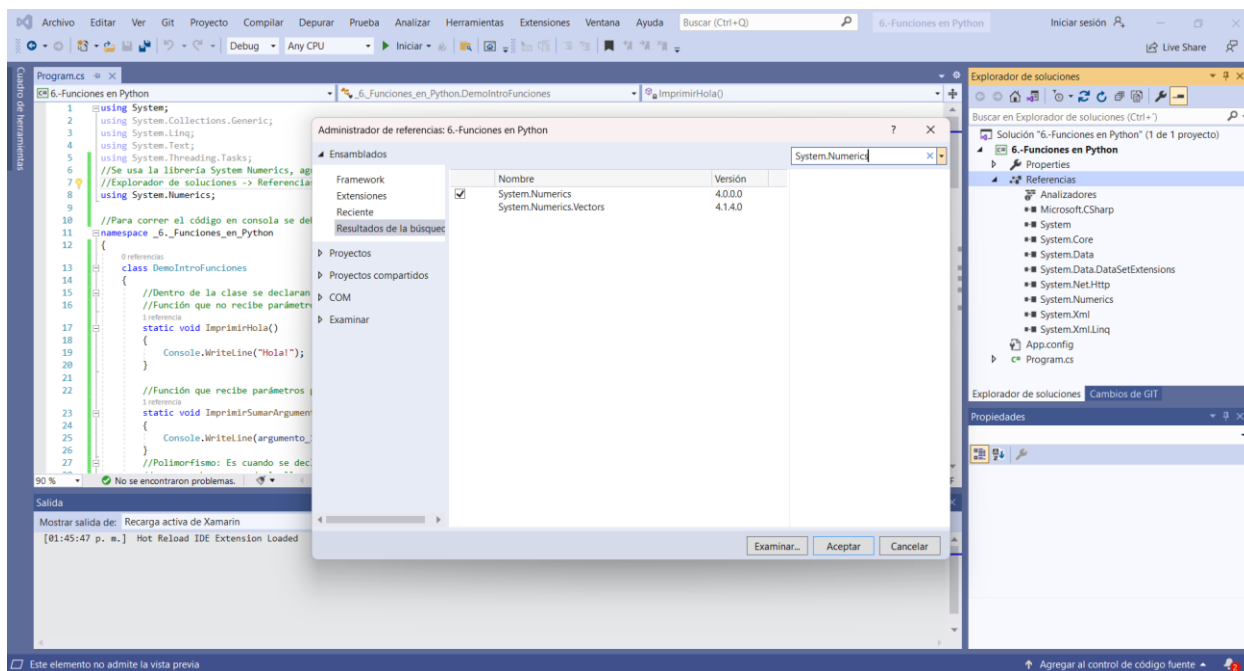
## Agregar Librerías a C# en Visual Studio en Todos los tipos de Plantillas de Visual Studio:

Para agregar cualquier librería a un programa hecho en C# con Visual Studio, no es suficiente ponerlo en el código, para ello deberemos dar clic derecho sobre la opción de: Explorador de Soluciones → Referencias → Agregar Referencia → Nombre Librería.



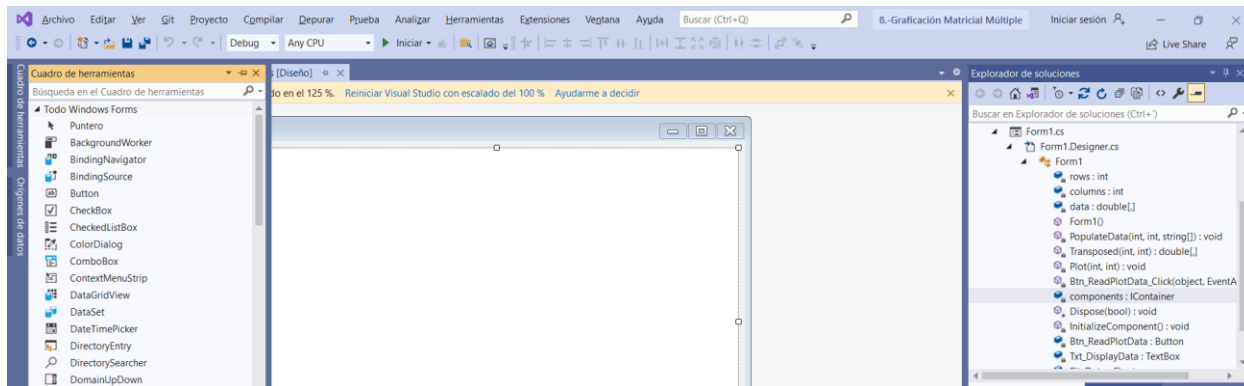
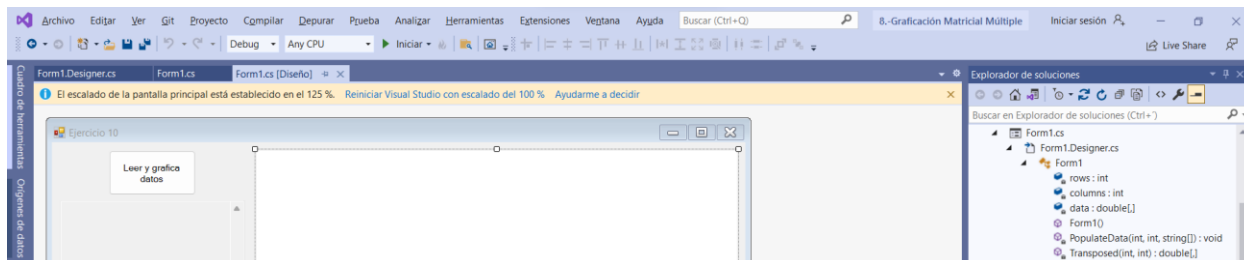
Al hacerlo aparecerá una ventana emergente en la cual debo buscar la librería que quiero agregar, una vez que la haya encontrado daré clic en el botón de Aceptar.

De esta manera ya se podrá utilizar esa librería en el programa de C#.



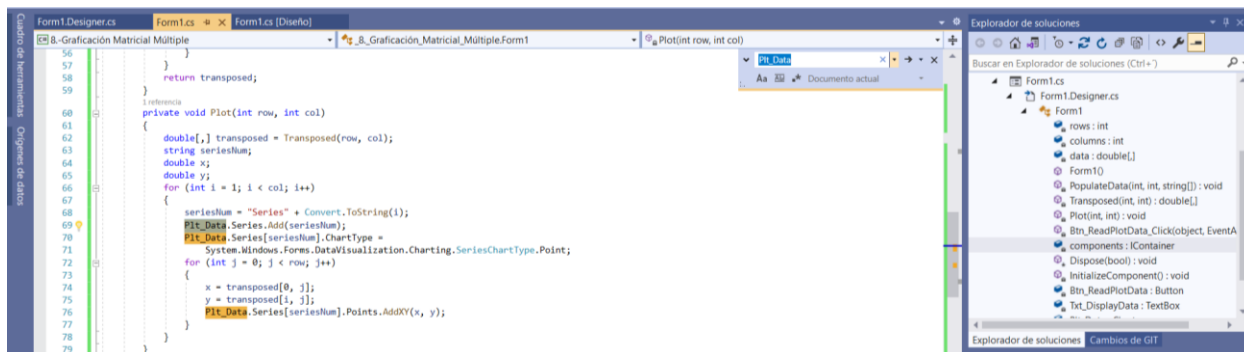
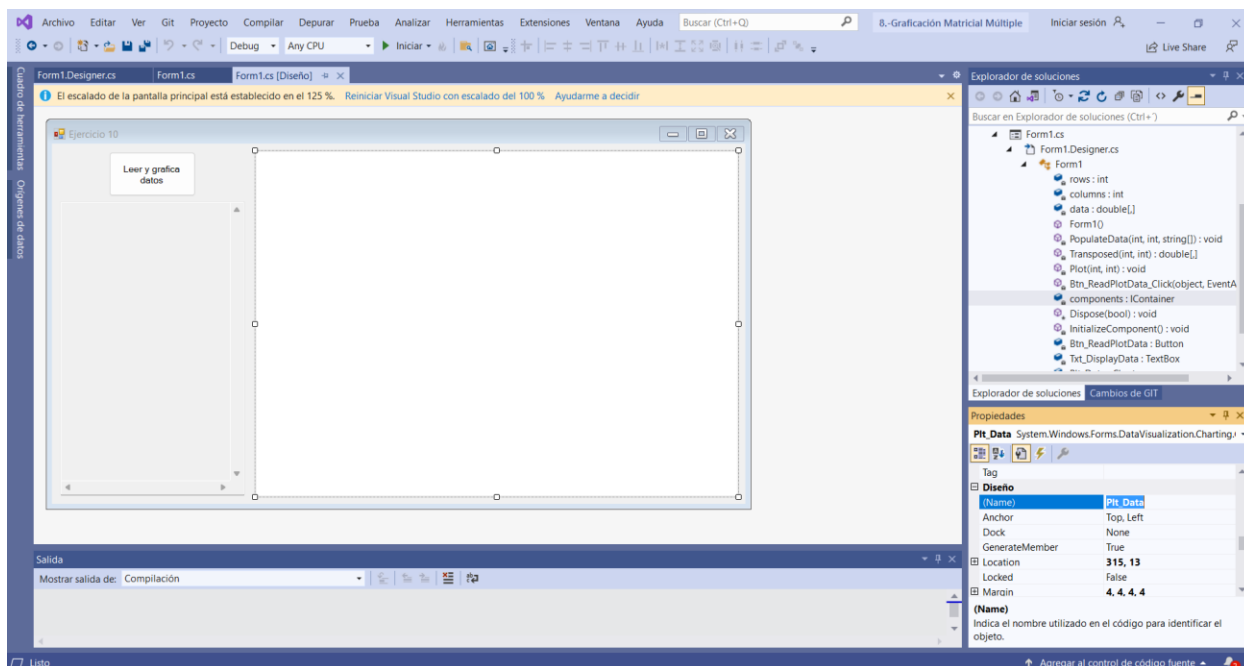
## Creación de GUI en Visual Studio Aplicación de Windows Forms (.NET Framework):

Al abrir un nuevo archivo de interfaz gráfica en Visual Studio (llamado Form), para poder agregar widgets (elementos gráficos) a la ventana, debemos situarnos en la pestaña de **Form.cs [Design]** para visualizar la nueva interfaz gráfica creada, luego debemos dar clic en la siguiente opción: Cuadro de Herramientas → Seleccionar el Widget que quiero añadir → Arrastrarlo Hacia el Form.





Posteriormente al dar clic sobre cada widget, en la esquina inferior derecha aparecerá la ventana de Properties, en donde deberé indicar el nombre y otros aspectos del widget para que por medio del código **Form.cs** pueda darles funcionalidad.



En el archivo de Form.Designer.cs se encontrarán codificados todos los elementos gráficos que hayamos añadido a la interfaz gráfica, incluidas sus propiedades como nombre, posición, tamaño, etc.

