**Instituto Tecnológico Nacional de México.**

**Instituto Tecnológico del Sur de Nayarit.**

**Trabajo:** Investigación.

**Maestra:** Cinthia Anahí Mata Bravo.

**Materia:** Programación orientada a objetos.

**Especialidad:** Itic´s

**Semestre:** 2

**Hecho por:**

**Angel Alejandro Parra Aguayo.**

**No. Control: 191140012**



[Aplicación de consola C# 3](#_Toc31748515)

[¿Qué es un proyecto de consola en c#? 3](#_Toc31748516)

[¿Cuáles son los archivos que se crean en consola c#? 3](#_Toc31748517)

[Apartado Properties 3](#_Toc31748518)

[Apartado Referencias. 4](#_Toc31748519)

[Archivo App.config 4](#_Toc31748520)

[Apartado Program.cs 5](#_Toc31748521)

[Aplicación Windowsform 5](#_Toc31748522)

[¿Qué es el windowsform en c#? 5](#_Toc31748523)

[¿Cuáles son los archivos que se crean en Windowsform? 5](#_Toc31748524)

[Apartado Properties 6](#_Toc31748525)

[Apartado Referencias 7](#_Toc31748526)

[App.config 7](#_Toc31748527)

[Form1.cs 8](#_Toc31748528)

[Program.cs 8](#_Toc31748529)

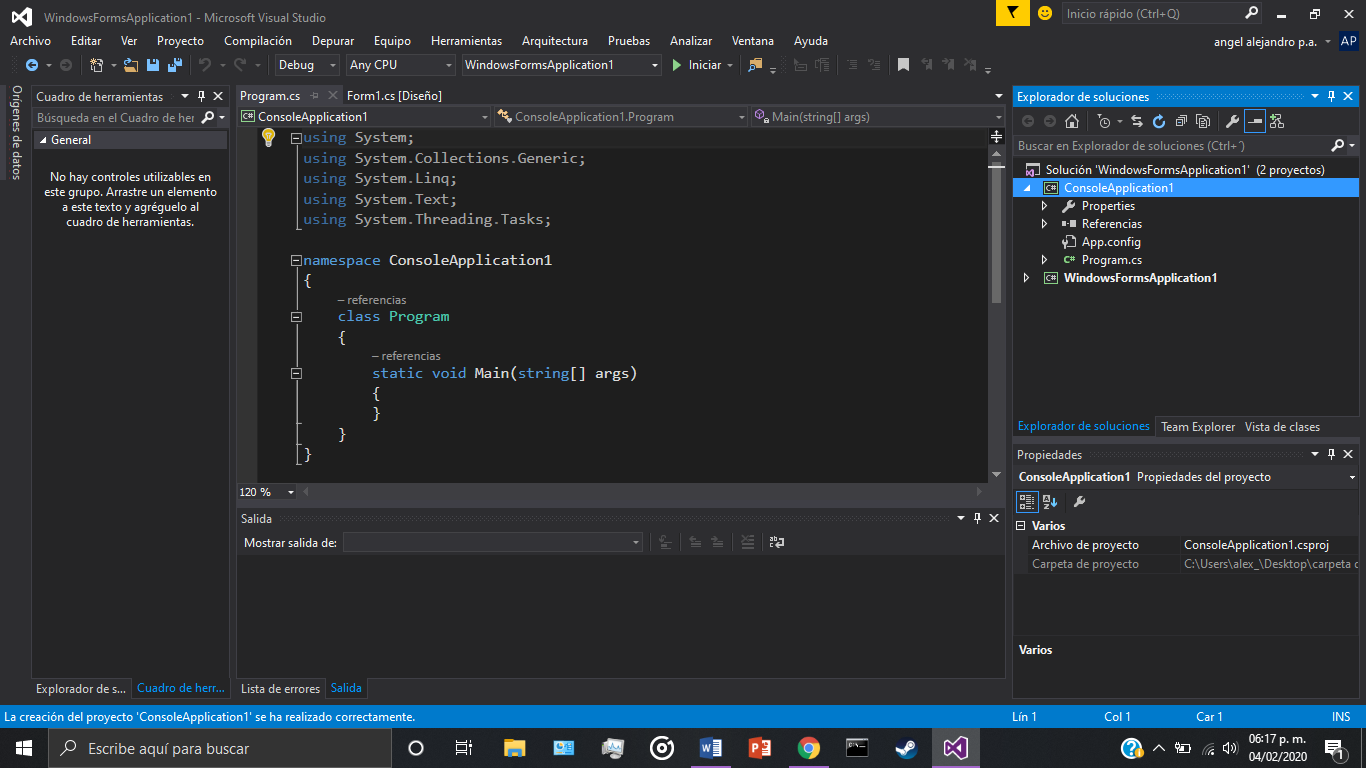
[Conclusión 9](#_Toc31748530)

# Aplicación de consola C#

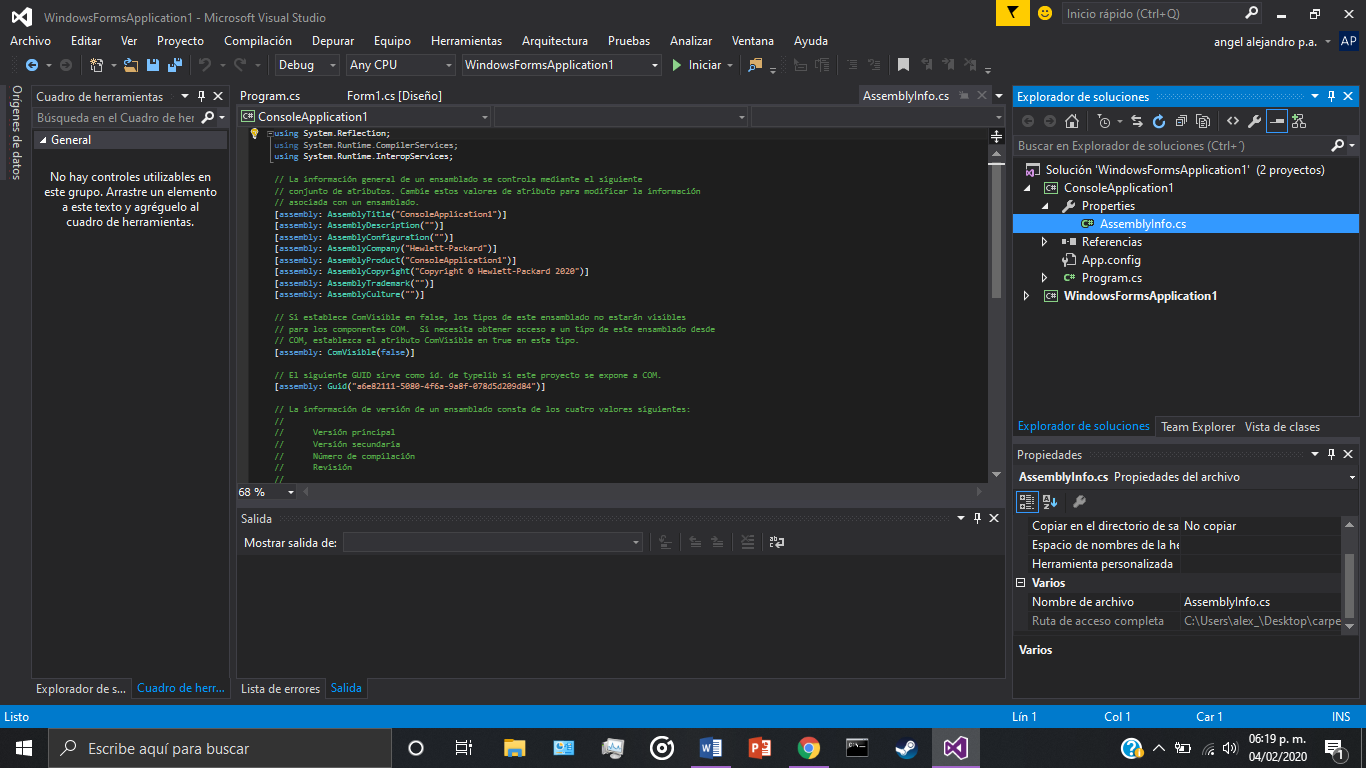
## ¿Qué es un proyecto de consola en c#?

Se puede definir una aplicación de consola como aquella que se ejecuta en una ventana de MS-DOS, es decir, en línea de comandos. Lo más común dentro del desarrollo bajo la plataforma .Net es la creación de aplicaciones Web o aplicaciones Windows sin embargo la mejor forma de sentar unas bases firmes acerca de la programación orientada a objetos es comenzar construyendo aplicaciones sencillas de consola.

## ¿Cuáles son los archivos que se crean en consola c#?

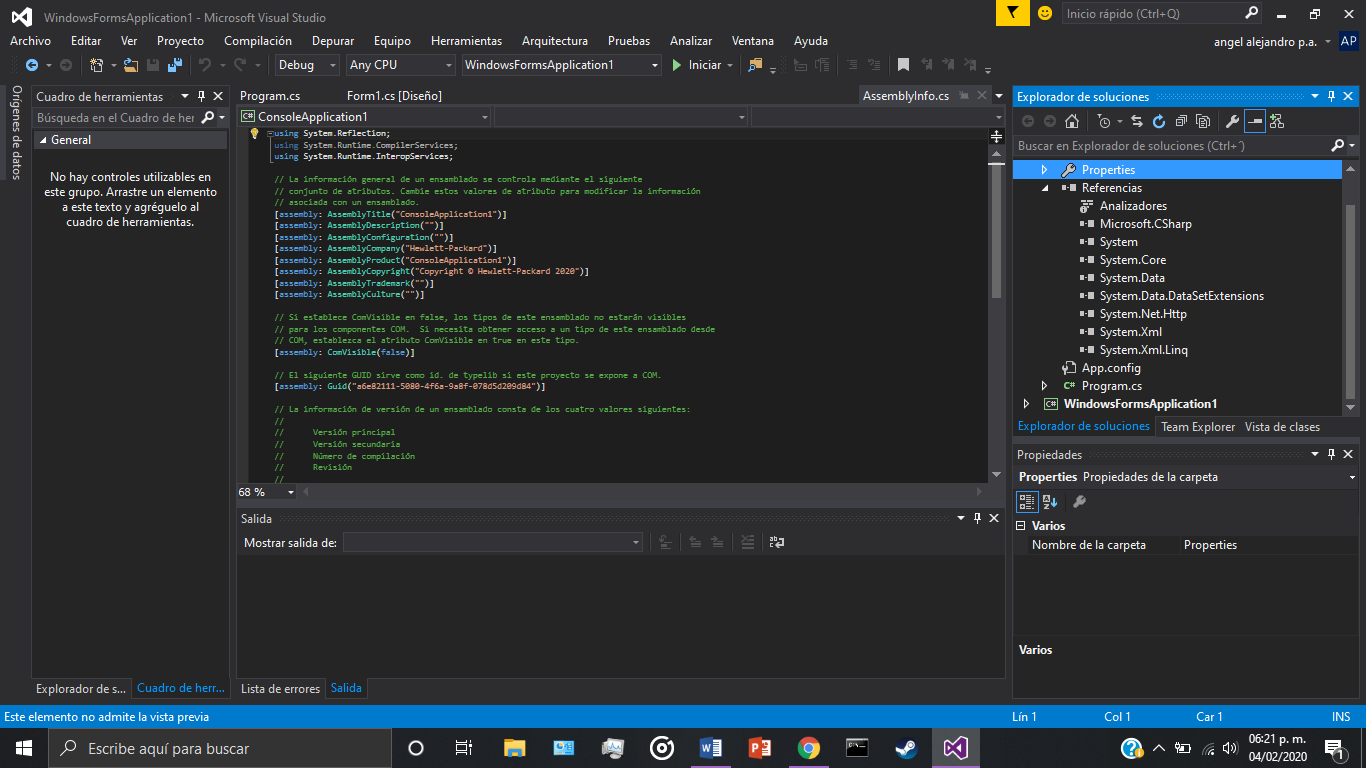
Al empezar el proyecto salen varios paquetes de inicio abajo del nombre del proyecto aparecen properties, referencias, app.config, program.cs y veremos sus partes.

# Apartado Properties

Al desplegar el apartadosale un nuevo archivo el cual se llama AssemblyInfo.cs y se usa por convención como el archivo de origen donde los desarrolladores colocan atributos de metadatos que describen el ensamblaje completo que están construyendo. Y con ese archivo se termina el apartado properties.

# Apartado Referencias.

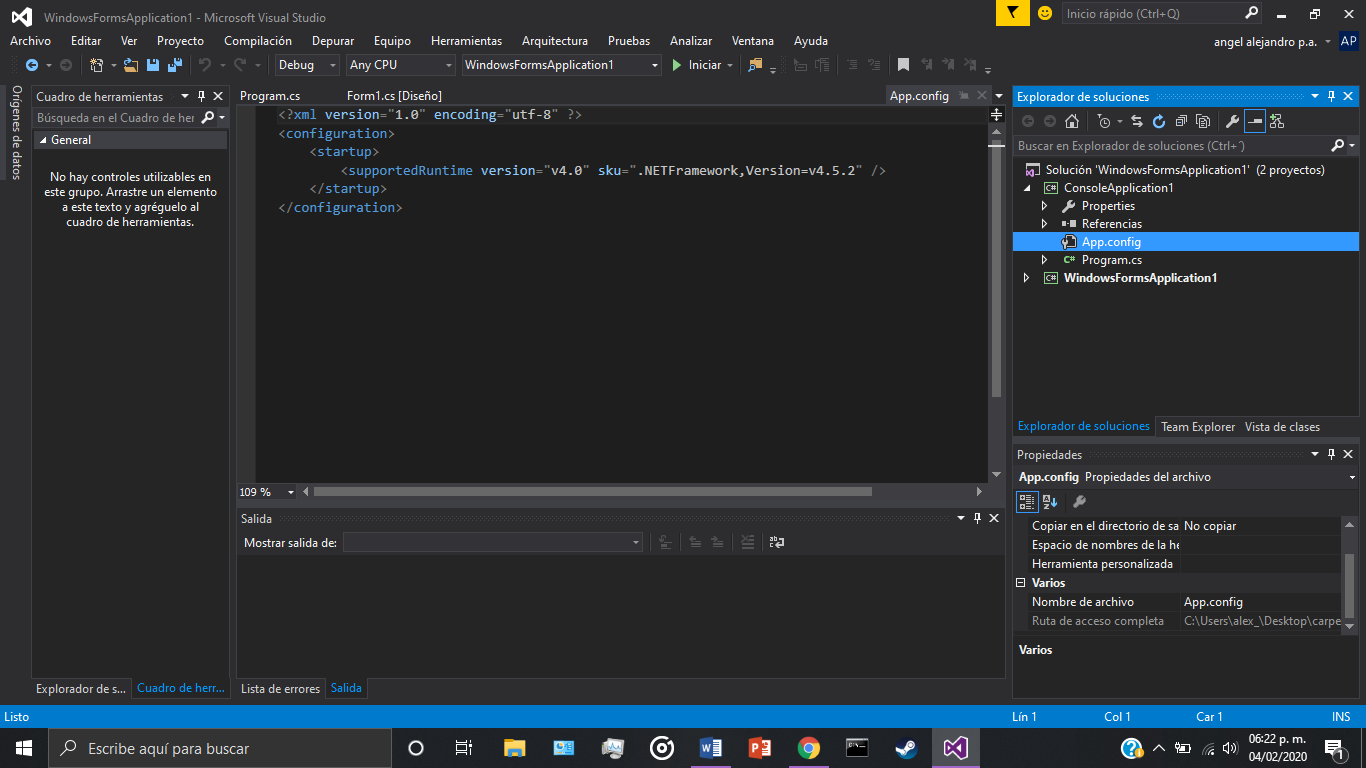
Este apartado no se toma mucho en cuenta ya que este apartado se usa para que funcione el mismo compilador y pueda seguir en varios aspectos.



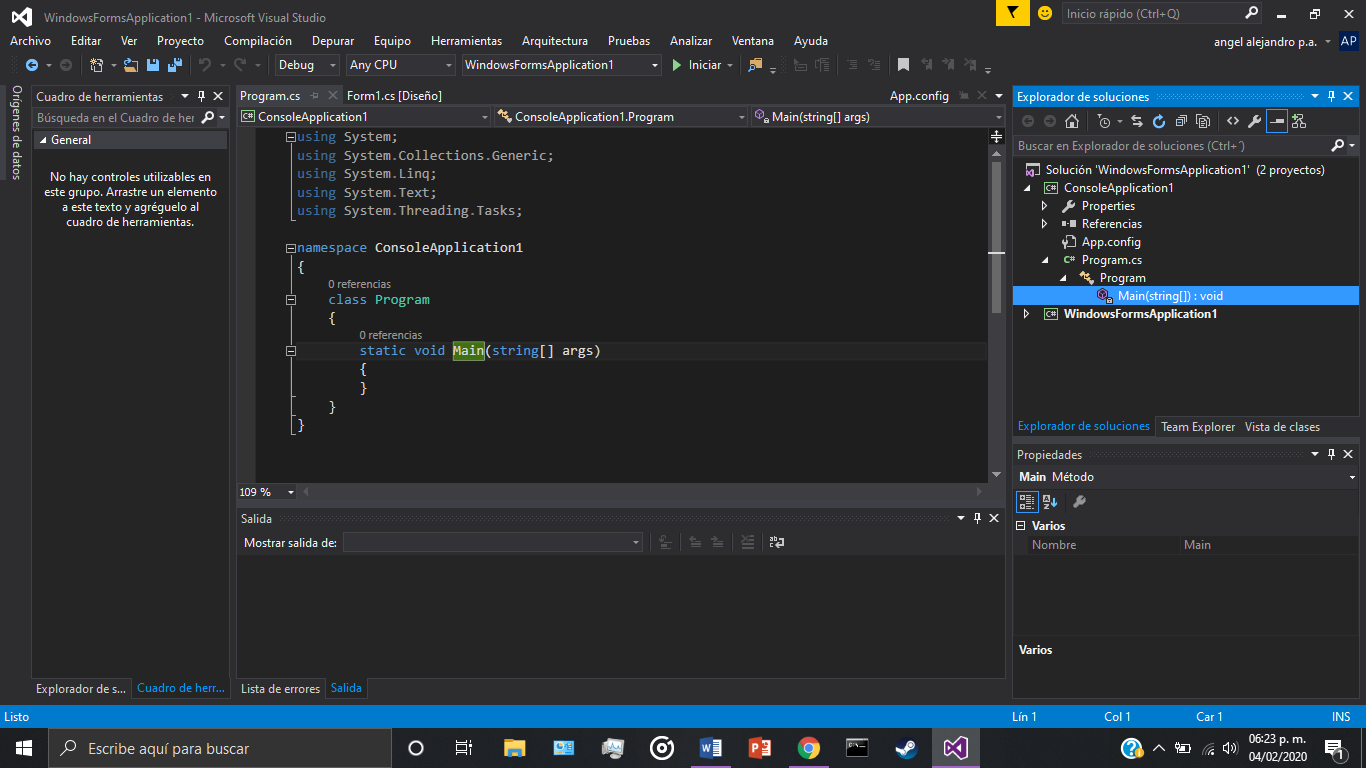
# Archivo App.config

El archivo de configuración de la aplicación (archivo app.config) a un proyecto de C #, puede personalizar la forma en que Common Language Runtime ubica y carga los archivos de ensamblaje.

Esto nos ayuda a programar con más precisión.



# Apartado Program.cs

En este apartado se puede observar al desplegar otra carpeta con el nombre de program el cual tiene otro archivo el cual recibe los valores del programa y los hace retribuibles haci que no se pueden regresar esos valores, program.cs es el principal de la aplicación. Esto se ejecutará primero cuando se ejecute la aplicación.

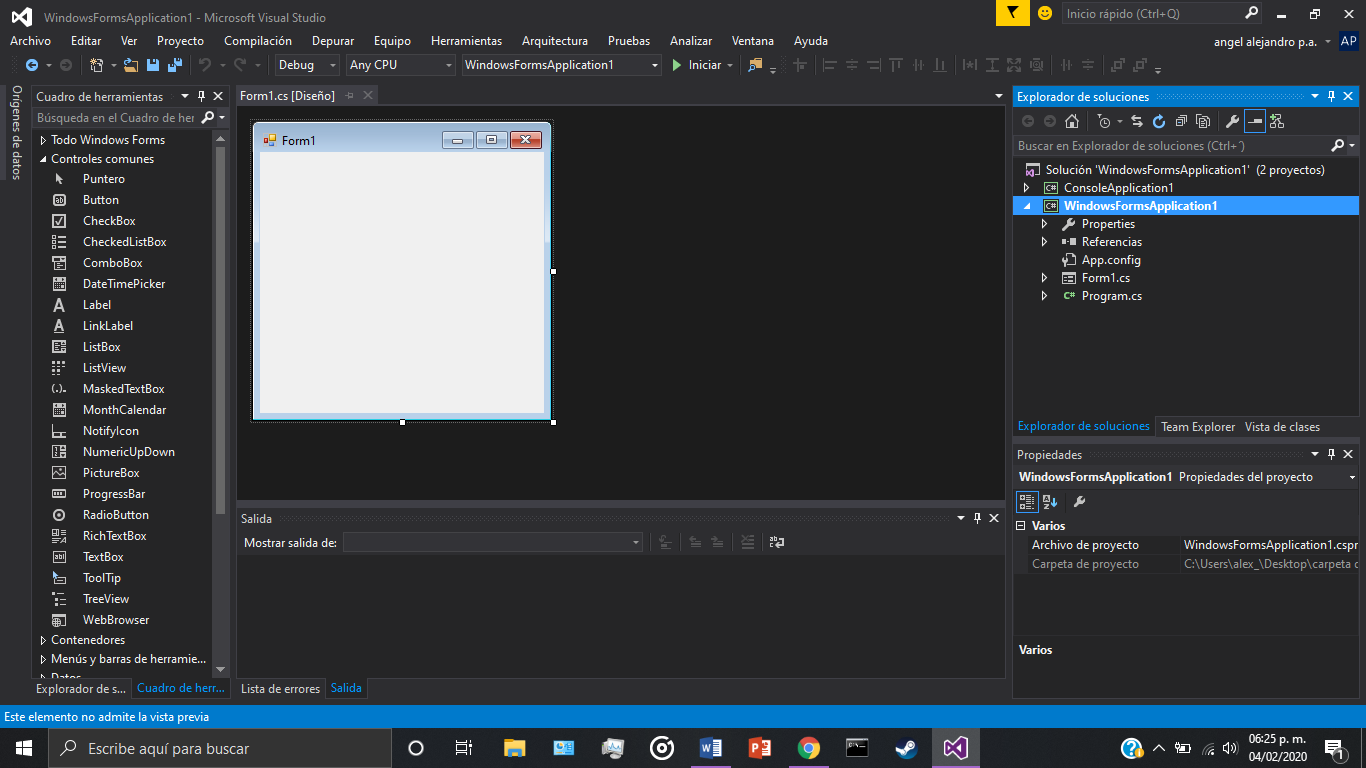
# Aplicación Windowsform

## ¿Qué es el windowsform en c#?

Como los formularios son la unidad básica de la aplicación, es esencial realizar algunas consideraciones sobre su función y diseño. En última instancia, un formulario es una pizarra en blanco que usted, como desarrollador, mejora con controles para crear una interfaz de usuario y con código para manipular los datos. Para ello, Visual Studio proporciona un entorno de desarrollo integrado (IDE) que ayuda a escribir código, así como un enriquecido conjunto de controles escrito con .NET Framework. Complementando la funcionalidad de estos controles con el código, puede desarrollar las soluciones que necesita de forma fácil y rápida.

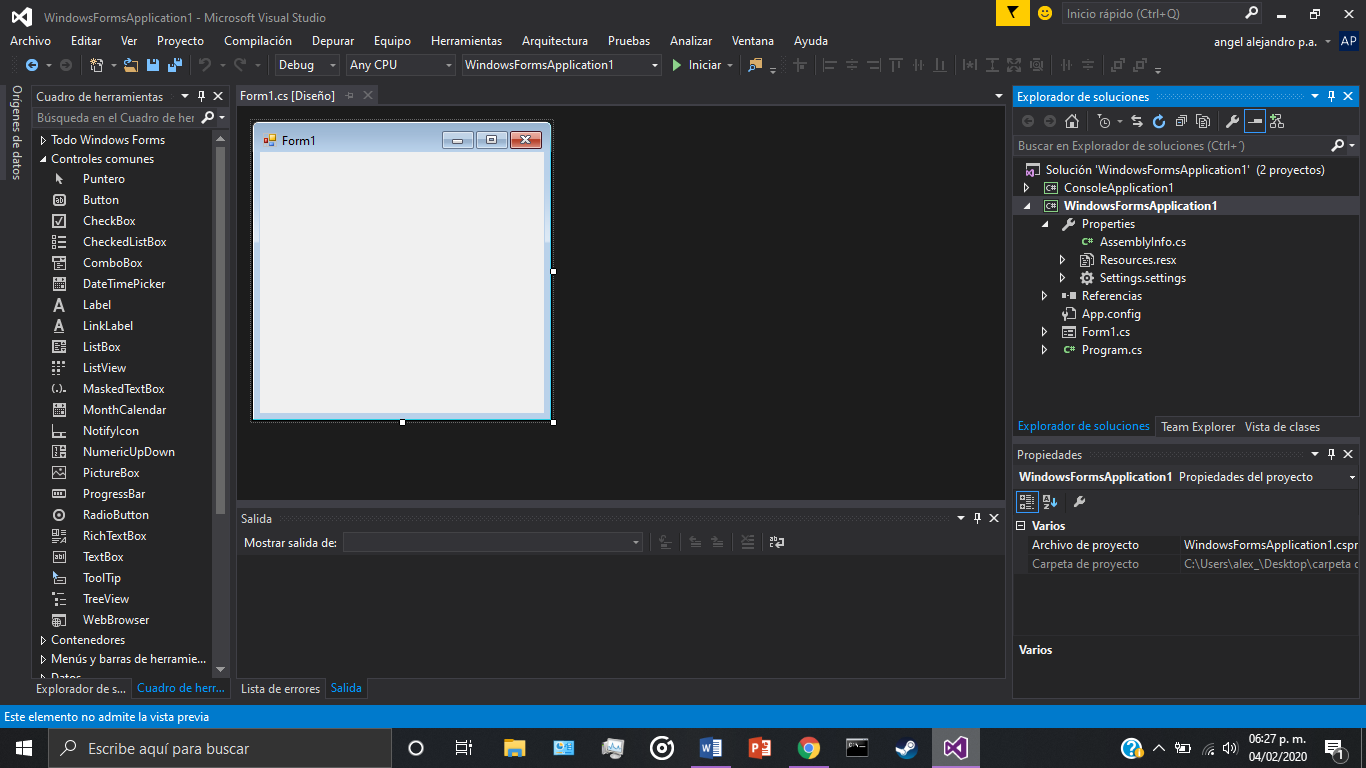
## ¿Cuáles son los archivos que se crean en Windowsform?

Los archivos que se crean al hacer un nuevo proyecto son la carpeta Properties donde adentro viene el archivo Assembly.info.cs, reources.resx, settings.settigns, referenias, app.config, form1.cs y program.cs



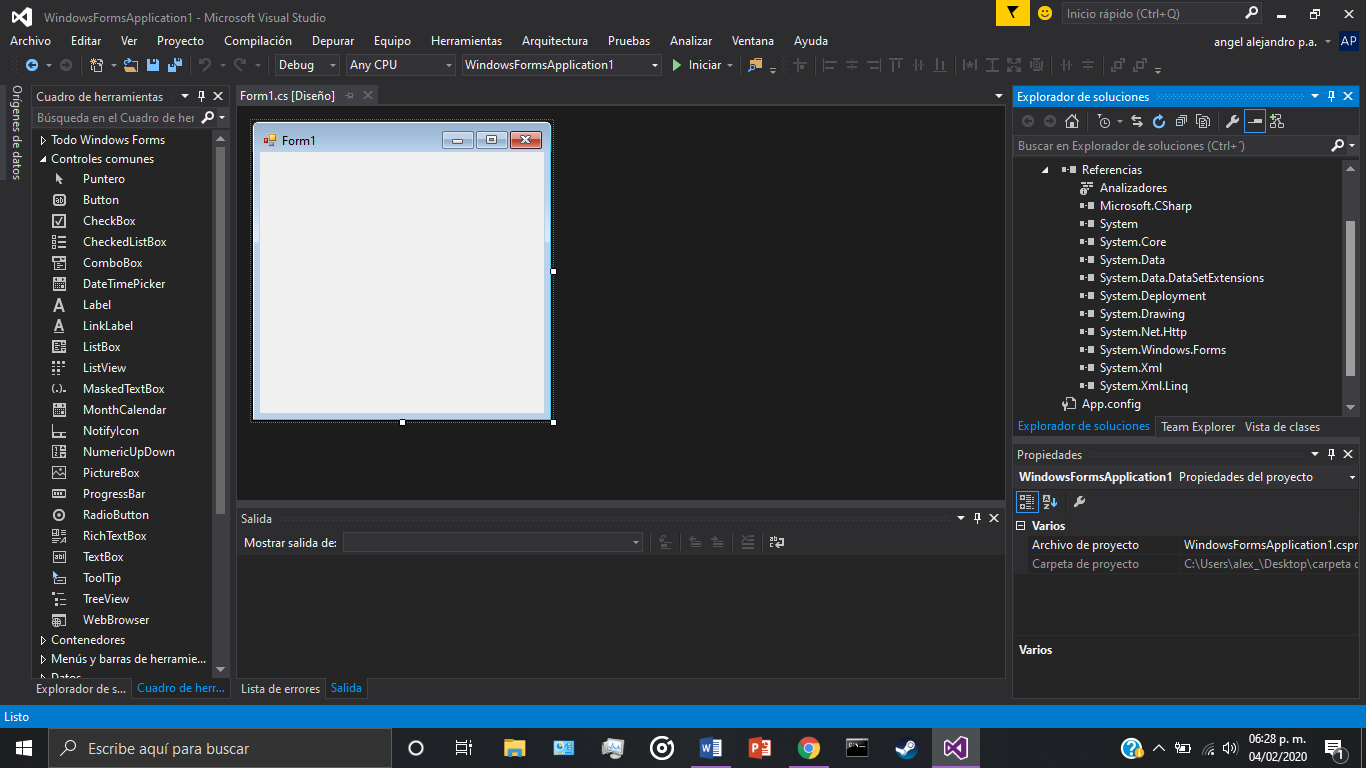
# Apartado Properties

Un control de Windows Forms hereda muchas propiedades de la clase base System.Windows.Forms.Control . Estos incluyen propiedades como Fuente, ForeColor, BackColor, Bounds, ClientRectangle, DisplayRectangle, Enabled, Focused, Height, Width, Visible, AutoSize y muchos otros. Para obtener detalles sobre las propiedades heredadas, consulte System.Windows. Puede anular las propiedades heredadas en su control, así como definir nuevas propiedades.



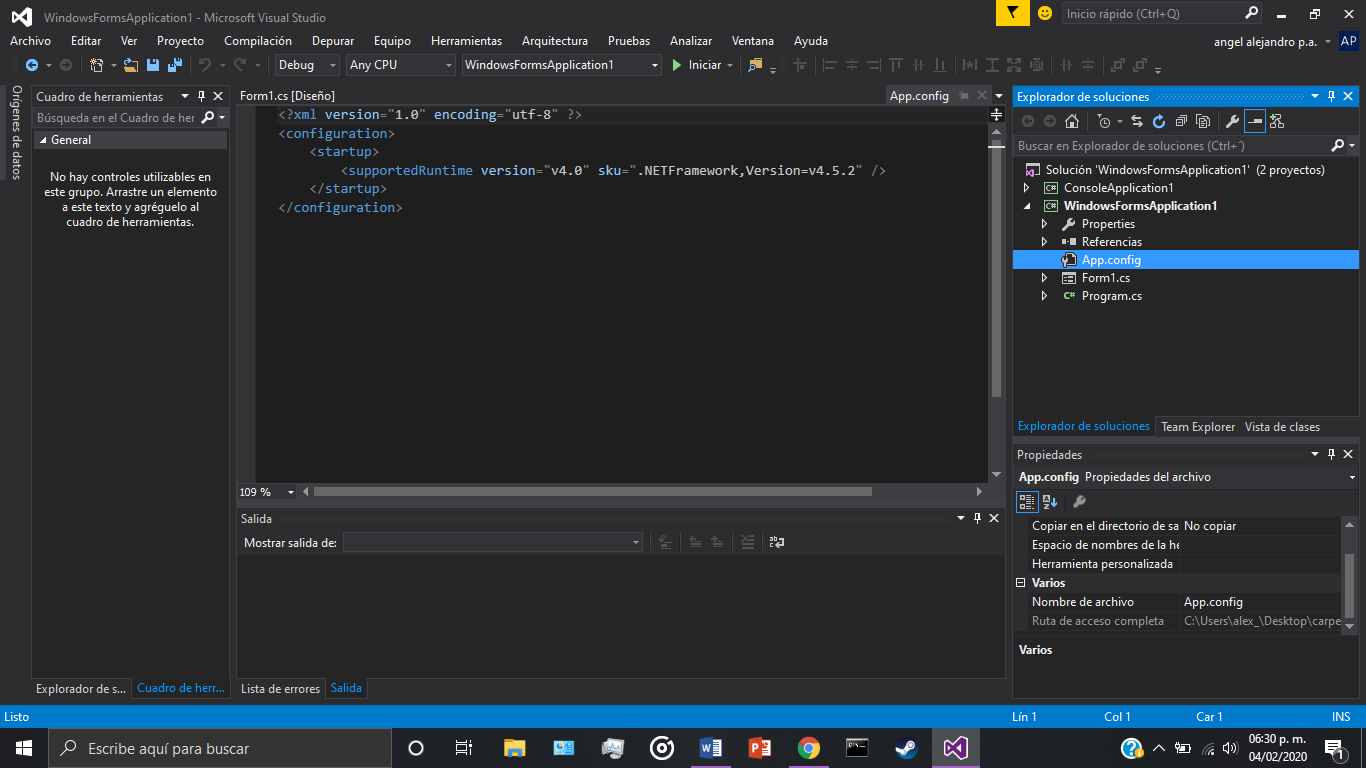
# Apartado Referencias

Antes de escribir código en un componente externo o en un servicio conectado, el proyecto debe contener primero una referencia a él. Una referencia es básicamente una entrada de un archivo de proyecto que contiene la información que Visual Studio necesita para localizar el componente o el servicio. Para agregar una referencia, haga clic con el botón derecho en el nodo Referencias o Dependencias del Explorador de soluciones y elija Agregar referencia. También puede hacer clic con el botón derecho en el nodo del proyecto y seleccionar Agregar > Referencia.



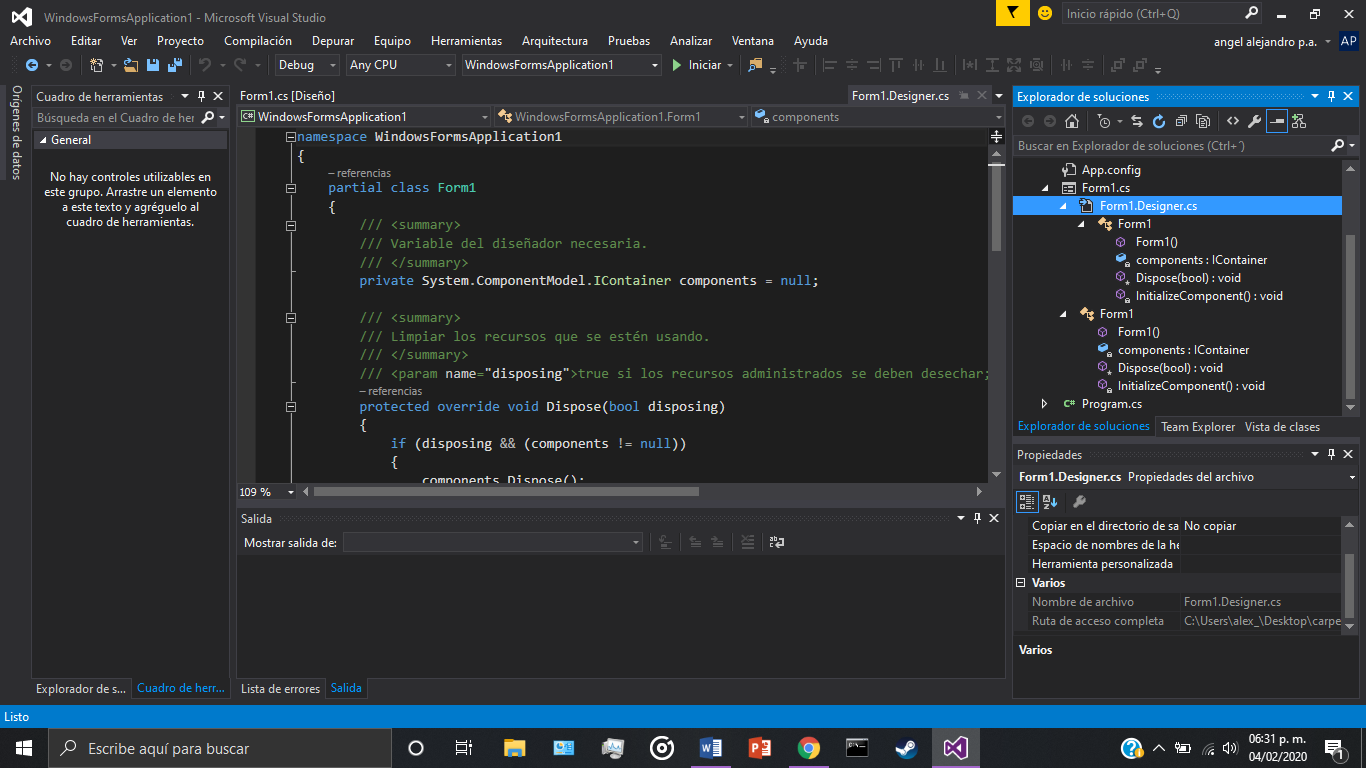
# App.config

Al agregar un archivo de configuración de la aplicación (archivo app.config) a un proyecto de C#, puede personalizar el modo en que Common Language Runtime busca y carga archivos de ensamblado. Para obtener más información sobre los archivos de configuración de la aplicación, vea cómo el motor en tiempo de ejecución ubica ensamblados.

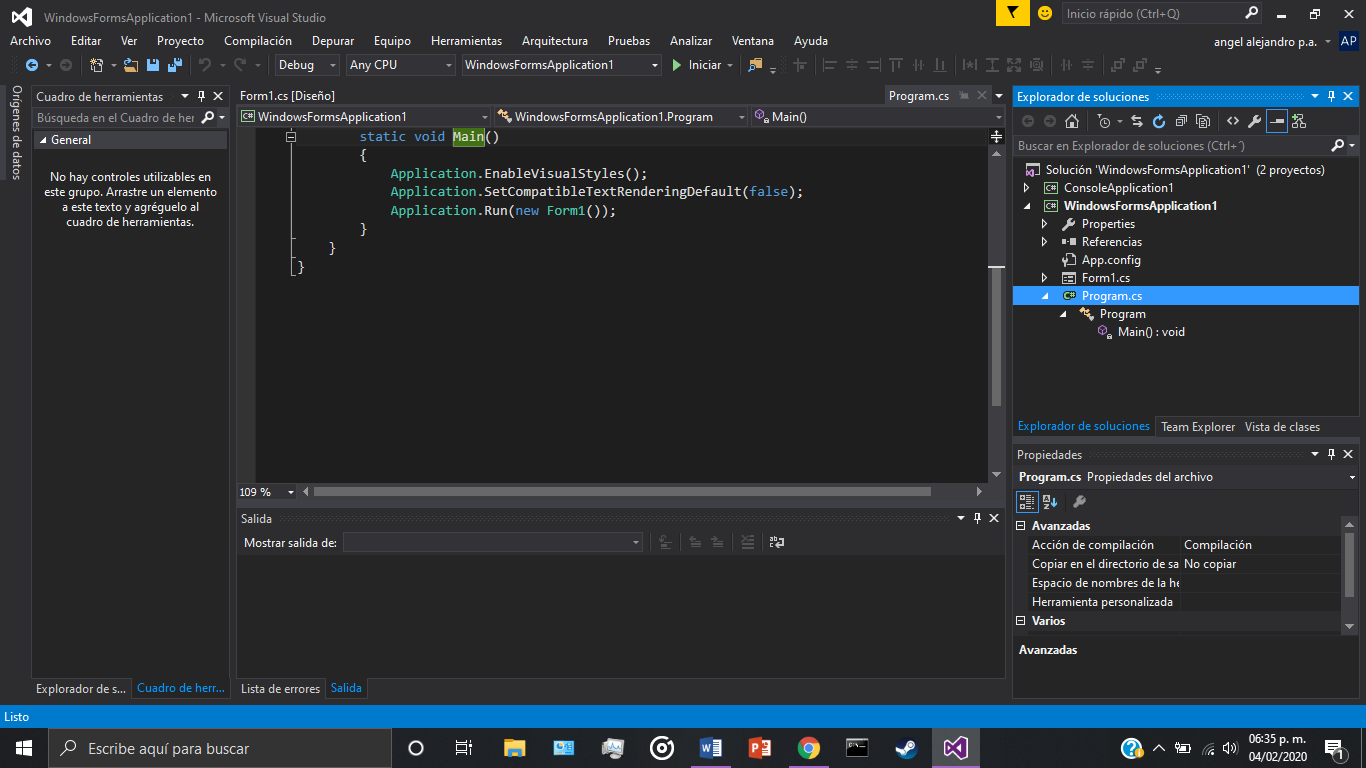


# Form1.cs

Form1.cs es el archivo de código subyacente del formulario de Windows. Es el archivo de clase del formulario de Windows donde se escriben los métodos necesarios, las funciones y los métodos y códigos controlados por eventos.

form1.designer.cs es el archivo del diseñador donde se inicializan los elementos del formulario. Si algún elemento se arrastra y suelta en la ventana del formulario, ese elemento se inicializará automáticamente en esta clase.

# Program.cs

program.cs es el principal de la aplicación. Esto se ejecutará primero cuando se ejecute la aplicación.

# Conclusión

En conclusión, podemos observar mejor cómo funciona la codificación en lenguaje windowsform y consola en el lenguaje de c# el cual se parece mucho al lenguaje que vimos el semestre anterior y combinado con java para crear las pantallas de una computadora, lo que diferencia de un lenguaje a otro es que uno solo aparece una pantalla negra y el otro es más visual y tiene más opciones para personalizarlo y pueda verse más estético. La necesidad de este semestre es comprender más este lenguaje de programación.