



**Universidad Autónoma de Chiapas**  
**Facultad de Contaduría y Administración, Campus I**  
**Licenciatura en Ingeniería y Desarrollo de Tecnologías de Software**

**Actividad: Visual .NET**

**Alumno: Georgina Andrea Méndez Hernández.**

**5º M.**

**Materia: Teoría matemática de la computación.**

**Docente: Juan José Tevera Mandujano.**

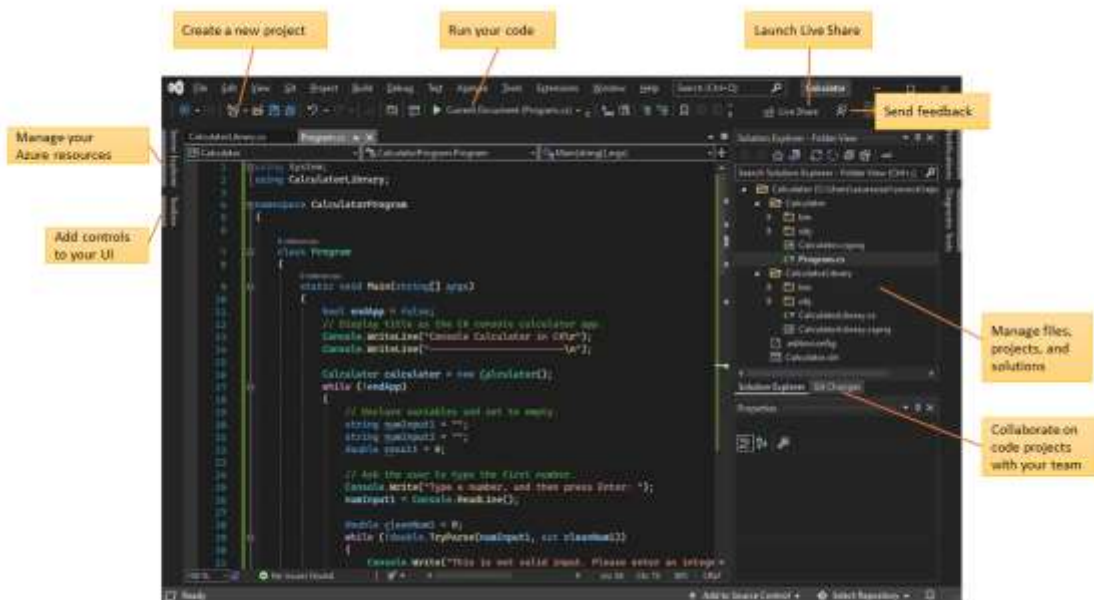
**Matricula: A220001.**

**Tuxtla Gutiérrez Chiapas.**

**05 de febrero de 2023.**

## ¿Qué es Visual Studio?

El IDE de Visual Studio es un panel de inicio creativo que se puede usar para editar, depurar y compilar código y, después, publicar una aplicación. Aparte del editor y el depurador estándar que proporcionan la mayoría de IDE, Visual Studio incluye compiladores, herramientas de finalización de código, diseñadores gráficos y muchas más características para facilitar el proceso de desarrollo de software.



En la imagen anterior se muestra Visual Studio con un proyecto abierto con las ventanas principales y su funcionalidad:

En la parte superior derecha del Explorador de soluciones, puede ver y administrar los archivos de código y navegar por ellos. El Explorador de soluciones puede ayudar a organizar el código al agrupar los archivos en soluciones y proyectos.

La ventana del editor central, donde es probable que pase la mayor parte del tiempo, muestra el contenido del archivo. En ella, podrá editar código o diseñar una interfaz de usuario, como una ventana con botones y cuadros de texto.

## Características

### Instalación modular

En el instalador modular de Visual Studio, elegirá e instalará las cargas de trabajo que desee. Las cargas de trabajo son grupos de características que los lenguajes de programación o las plataformas necesitan para funcionar. Esta estrategia

modular ayuda a reducir la superficie de instalación de Visual Studio, por lo que se instala y actualiza más rápido.

## Creación de aplicaciones de Azure habilitadas para la nube

Visual Studio ofrece un conjunto de herramientas para crear fácilmente aplicaciones habilitadas para la nube de Microsoft Azure.

## Creación de aplicaciones web

Visual Studio puede ayudarle a escribir aplicaciones para la Web. Puede crear aplicaciones web mediante ASP.NET, Node.js, Python, JavaScript y TypeScript. Visual Studio admite muchos marcos web, como Angular, jQuery y Express.

ASP.NET Core y .NET Core funcionan en los sistemas operativos Windows, Mac y Linux. ASP.NET Core es una actualización principal a MVC, WebAPI y SignalR. ASP.NET Core se diseñó desde la base para ofrecer una pila de .NET eficiente y componible, con el fin de compilar servicios y aplicaciones web modernos basados en la nube.

## Compilar aplicaciones y juegos multiplataforma

Visual Studio puede crear aplicaciones y juegos para macOS, Linux y Windows, así como para Android, iOS y otros dispositivos móviles. Con Visual Studio, puede crear:

Aplicaciones de .NET Core que se ejecutan en Windows, macOS y Linux.

Aplicaciones móviles para iOS, Android y Windows en C# y F# mediante Xamarin.

Juegos 2D y 3D en C# mediante Visual Studio Tools para Unity.

Aplicaciones de C++ nativas para dispositivos iOS, Android y Windows. Comparta código común en bibliotecas para iOS, Android y Windows mediante C++ para desarrollo multiplataforma.

## Conectarse a bases de datos

El Explorador de servidores ayuda a explorar y administrar instancias y recursos de servidor de forma local y remota, y en Azure, Microsoft 365, Salesforce.com y sitios web.

El Explorador de objetos de SQL Server ofrece una vista de los objetos de base de datos similar a la de SQL Server Management Studio. Con el Explorador de objetos de SQL Server puede realizar trabajos de administración y diseño de bases de datos ligeras. Algunos ejemplos son la edición de datos de tabla, la comparación de esquemas y la ejecución de consultas mediante menús contextuales.

SQL Server Data Tools (SSDT) es un eficaz entorno de desarrollo para SQL Server, Azure SQL Database y Azure SQL Data Warehouse. Con SSDT, puede compilar, depurar, mantener y refactorizar bases de datos. Puede trabajar con un proyecto de

base de datos o directamente con una instancia de base de datos conectada de manera local o externa.

Depure, pruebe y mejore su código

Cuando escribe código, debe ejecutarlo y probarlo para comprobar que funciona sin errores. Con el sistema de depuración de Visual Studio, puede depurar el código que se ejecuta en el proyecto local, en un dispositivo remoto o en un emulador de dispositivo. Ejecute el código una instrucción cada vez e inspeccione las variables mientras avanza. O bien, establezca puntos de interrupción que solo se alcanzan cuando una condición especificada es verdadera.

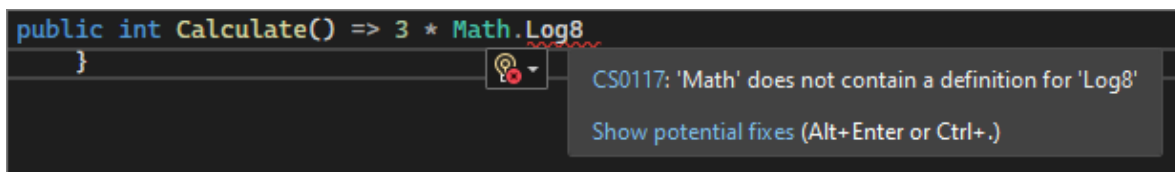
Administrar el código fuente y colaborar con otras personas

En Visual Studio, puede administrar el código fuente en los repositorios de Git hospedados por cualquier proveedor, incluido GitHub.

## **Características de productividad populares**

Subrayados ondulados y Acciones rápidas

Los subrayados ondulados son rayas con formas de onda debajo de las palabras que alertan de errores o posibles problemas en el código a medida que se escribe. Estas pistas visuales le ayudan a corregir problemas inmediatamente, sin esperar a detectar errores durante la compilación o el tiempo de ejecución. Si mantiene el puntero sobre un subrayado rojo, verá más información sobre el error. También puede aparecer una bombilla en el margen izquierdo que muestra acciones rápidas que puede realizar para corregir el error.



Limpieza de código

Con solo el clic de un botón, puede dar formato al código y aplicar cualquier corrección de código sugerida por la configuración del estilo del código, las convenciones de. `editorconfig` y los analizadores de Roslyn. La limpieza de código, disponible actualmente solo para el código de C#, ayuda a solucionar problemas con el código antes de que pase a revisión.

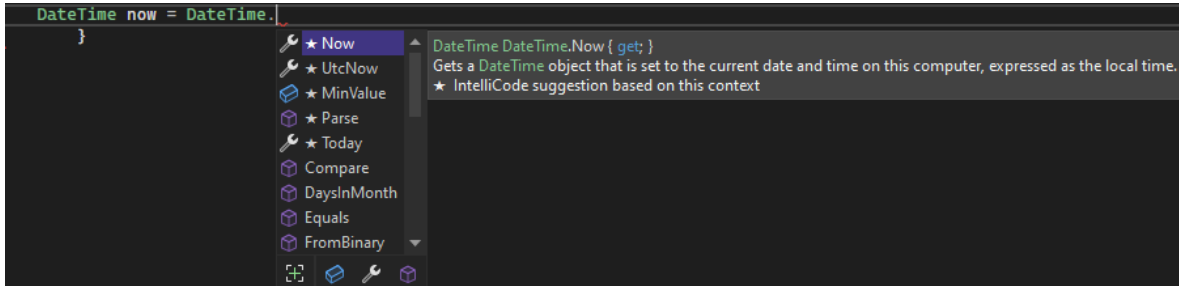
Refactorización

La refactorización incluye operaciones como el cambio de nombre inteligente de variables, la extracción de una o más líneas de código en un nuevo método y el cambio del orden de los parámetros de método.

## IntelliSense

IntelliSense es un conjunto de características que muestran información sobre el código directamente en el editor y, en algunos casos, escriben pequeños fragmentos de código automáticamente. Es como tener documentación básica insertada en el editor, lo que evita tener que buscar información escrita en cualquier otro lugar.

La siguiente ilustración muestra cómo IntelliSense muestra una lista de miembros de un tipo:



## Búsqueda de Visual Studio

Los menús, las opciones y las propiedades de Visual Studio pueden parecer abrumadores en ocasiones. La búsqueda de Visual Studio, o Ctrl+Q, es una excelente manera de encontrar rápidamente las características del IDE y el código en un solo lugar.

## Live Share

Edita y depura en colaboración con otros usuarios en tiempo real, sin importar el tipo de aplicación o el lenguaje de programación. Puede compartir el proyecto al instante y de forma segura. También puede compartir sesiones de depuración, instancias de terminal, aplicaciones web de localhost, llamadas de voz, etc.

## Jerarquía de llamadas

En la ventana Jerarquía de llamadas se muestran los métodos que llaman a un método seleccionado. Esta información puede ser útil si está pensando en cambiar o quitar el método, o si está intentando realizar el seguimiento de un error.

## CodeLens

CodeLens ayuda a buscar referencias al código, cambios en él, errores vinculados, elementos de trabajo, revisiones de código y pruebas unitarias, sin salir del editor.

Ir a definición

La característica Ir a definición lleva directamente a la ubicación de una definición de tipo o función.

Ver la definición

En la ventana Ver la definición se muestra una definición de tipo o método sin abrir un archivo aparte.

Recarga activa

Recarga activa permite editar los archivos de código de la aplicación y aplicar los cambios de código inmediatamente a la aplicación en ejecución.

### **¿Qué es .NET framework, CLR, IDE?**

.NET Framework es un entorno de ejecución administrado para Windows que proporciona diversos servicios a las aplicaciones en ejecución. Consta de dos componentes principales: Common Language Runtime (CLR), que es el motor de ejecución que controla las aplicaciones en ejecución, y la biblioteca de clases de .NET Framework, que proporciona una biblioteca de código probado y reutilizable al que pueden llamar los desarrolladores desde sus propias aplicaciones. Los servicios que ofrece .NET Framework a las aplicaciones en ejecución son los siguientes:

Administración de la memoria. En muchos lenguajes de programación, los programadores son responsables de asignar y liberar memoria y de administrar la vida útil de los objetos. En las aplicaciones de .NET Framework, CLR proporciona estos servicios en nombre de la aplicación.

Sistema de tipos comunes. En los lenguajes de programación tradicionales, el compilador define los tipos básicos, lo que complica la interoperabilidad entre lenguajes. En .NET Framework, los tipos básicos los define el sistema de tipos de .NET Framework y son comunes a todos los lenguajes que tienen como destino .NET Framework.

Biblioteca de clases extensa. En lugar de tener que escribir cantidades extensas de código para controlar operaciones comunes de programación de bajo nivel, los programadores usan una biblioteca de tipos accesible en todo momento y sus miembros desde la biblioteca de clases de .NET Framework.

Interoperabilidad de lenguajes. Los compiladores de lenguajes cuya plataforma de destino es .NET Framework emiten un código intermedio denominado Lenguaje intermedio común (CIL), que, a su vez, se compila en tiempo de ejecución a través de Common Language Runtime. Con esta característica, las rutinas escritas en un

lenguaje son accesibles para otros lenguajes, de modo que los programadores puedan centrarse en crear aplicaciones en su lenguaje preferido.

**Ejecución en paralelo.** .NET Framework ayuda a resolver conflictos entre versiones y permite que varias versiones de Common Language Runtime coexistan en el mismo equipo. Esto significa que pueden coexistir varias versiones de las aplicaciones, y que una aplicación se puede ejecutar en la versión de .NET Framework con la que se compiló. **Compatibilidad con múltiples versiones (multi-targeting).** Al establecer .NET Standard como destino, los desarrolladores crear bibliotecas de clases que funcionan en varias plataformas de .NET Framework compatibles con esa versión del estándar.

## **CLR**

.NET proporciona un entorno en tiempo de ejecución denominado Common Language Runtime que ejecuta el código y proporciona servicios que facilitan el proceso de desarrollo.

Los compiladores y las herramientas exponen la funcionalidad de Common Language Runtime y permiten escribir código que se beneficia del entorno de ejecución administrado. El código que se desarrolla con un compilador de lenguaje y que tiene como destino el entorno de ejecución se denomina código administrado. El código administrado se beneficia de características como la integración entre lenguajes, el control de excepciones entre lenguajes, la seguridad mejorada, la compatibilidad con la implementación y el control de versiones, un modelo simplificado para interacción de componentes, y servicios de generación de perfiles y depuración.

Common Language Runtime facilita el diseño de los componentes y de las aplicaciones cuyos objetos interactúan entre lenguajes distintos. Los objetos escritos en lenguajes diferentes pueden comunicarse entre sí, lo que permite integrar sus comportamientos de forma precisa. Por ejemplo, puede definir una clase y, a continuación, utilizar un lenguaje diferente para derivar una clase de la clase original o llamar a un método de la clase original. También se puede pasar al método de una clase una instancia de una clase escrita en un lenguaje diferente. Esta integración entre lenguajes es posible porque los compiladores de lenguaje y las herramientas que se dirigen al tiempo de ejecución usan un sistema de tipos común definido por el runtime. Siguen las reglas del entorno de ejecución para definir nuevos tipos y para crear, usar, conservar y enlazar a tipos.

## **:NET core**

ASP.NET es un marco de desarrollo web popular para compilar aplicaciones web en la Plataforma .NET.

ASP.NET Core es la versión de código abierto de ASP.NET, que se ejecuta en macOS, Linux y Windows. ASP.NET Core se lanzó por primera vez en 2016 y es un rediseño de las versiones anteriores de ASP.NET solo para Windows.

ASP.NET Core está diseñado para permitir que los componentes en tiempo de ejecución, las API, los compiladores y los lenguajes evolucionen rápidamente, a la vez que proporciona una plataforma estable y compatible para mantener las aplicaciones en ejecución.

Pueden existir varias versiones de ASP.NET Core en paralelo en el mismo servidor. Lo que significa que una aplicación puede adoptar la versión más reciente, mientras que otras aplicaciones siguen ejecutándose en la versión en la que se probaron.

ASP.NET Core proporciona varias opciones de soporte técnico de ciclo de vida para satisfacer las necesidades de la aplicación.

### **Lenguajes nativos utilizados en Visual Studio**

- Python
- C++
- C#
- TypeScript
- Visual Basic
- F#
- JavaScript

#### **Fuentes**

Sacado de la página principal de MICROSOFT

<https://visualstudio.microsoft.com/es/vs/features/web/languages/>

<https://learn.microsoft.com/es-es/visualstudio/ide/advanced-feature-overview?view=vs-2022><https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022>

<https://dotnet.microsoft.com/es-es/learn/aspnet/what-is-aspnet-core>

<https://learn.microsoft.com/es-es/dotnet/framework/get-started/>