



Microsoft Visual

Studio

Índice

1. *¿Qué es Microsoft Visual Studio?*
2. *Características principales*
3. *Instalación y setting up*
4. *Configuración post-instalación*
5. *Funcionalidades principales*
6. *Extensiones y plugins*
7. *Ventajas y desventajas*
8. *Diferencias con Visual Studio Code*

¿Qué es Microsoft Visual Studio?

Se trata de un entorno de desarrollo integrado (IDE) desarrollado por Microsoft para sistemas operativos Windows y Mac. Caracterizado por la cantidad de funciones que apoyan el desarrollo de software, cómo escribir, editar, depurar y compilar el código y, luego, implementar la aplicación.

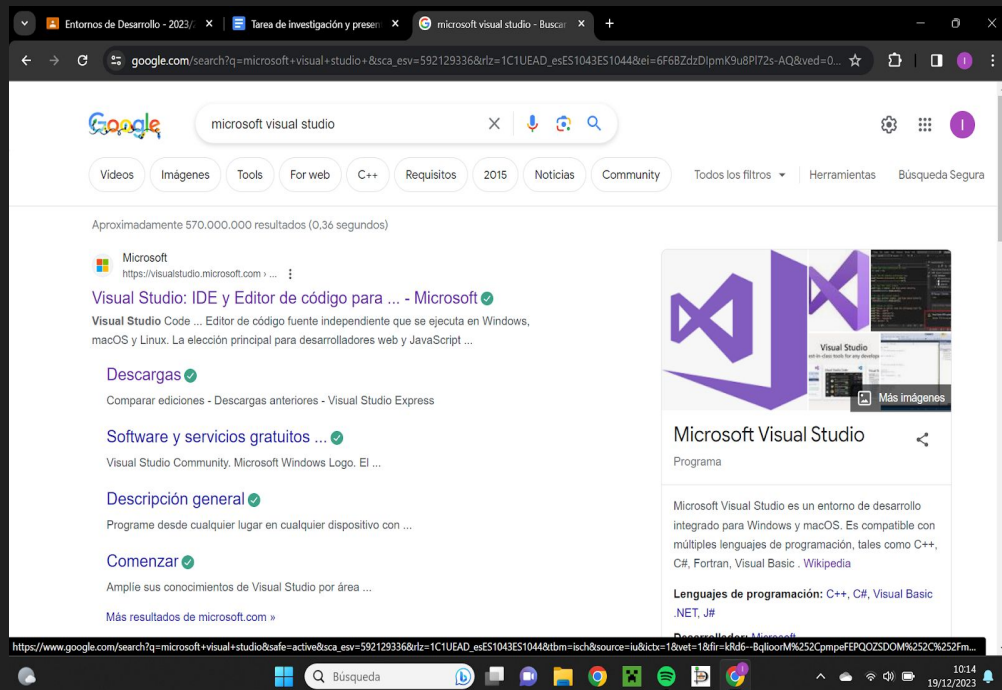
Características principales

- **Herramientas y características de codificación eficaces:** todo lo que necesita para compilar sus aplicaciones en un solo lugar.
- **Compatibilidad con varios lenguajes de programación:** C++, C#, JavaScript, Python, etc.
- **Desarrollo multiplataforma:** compilación de aplicaciones para cualquier plataforma.
- **Integración del control de versiones:** colaboración en el código con compañeros de equipo.
- **Desarrollo asistido por inteligencia artificial:** escritura de código de forma más eficaz con ayuda de la inteligencia artificial.

[Review completa en Youtube](#)

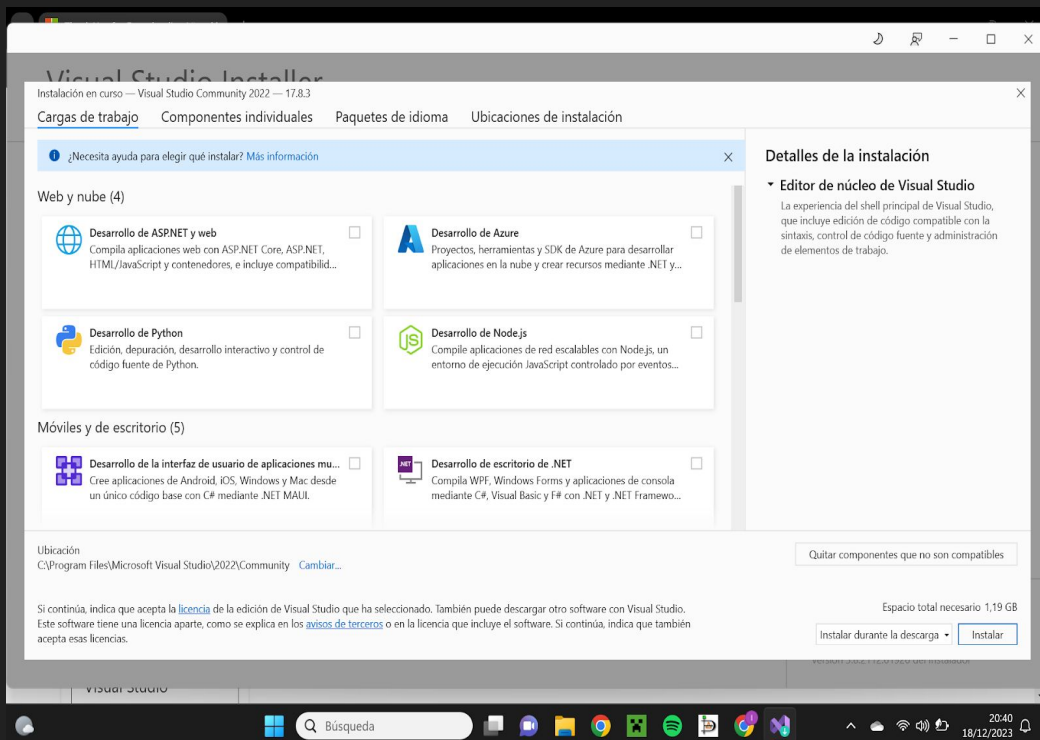
Instalación

Para comenzar con la instalación de Microsoft Visual Studio buscaremos el IDE en internet y entraremos al enlace que lleva a la página oficial de Microsoft, donde lo descargaremos.



Setting up

Tenemos la opción de descargar las cargas de trabajo que nosotros elijamos, según en qué ámbito nos vayamos a enfocar, mientras más escojamos, mayor será el almacenamiento que ocupe Visual. Tras elegir las cargas empezará la instalación final la cual notificará su finalización.



Configuración Post-Instalación de Visual Studio



1. **Iniciar Visual Studio:** Al abrirlo por primera vez, puede solicitarte iniciar sesión con una cuenta de Microsoft.
1. **Configuración del Entorno:** Personaliza el entorno según tus preferencias (temas visuales, configuraciones de editor, preferencias de teclado, etc.).
1. **Instalar Extensiones Adicionales:** Añade funcionalidades desde el Visual Studio Marketplace, accesible desde la opción "Extensiones" en el menú.
1. **Actualizaciones:** Mantén Visual Studio actualizado para aprovechar nuevas características y correcciones.

Configuración Post-Instalación de Visual Studio



Consejos Adicionales:

Requisitos del Sistema: Verifica los requisitos mínimos antes de instalar.

Instalaciones Paralelas: Es posible tener varias versiones de Visual Studio en el mismo sistema.

Configuración de Proyectos: Configura tus proyectos con plantillas específicas según el tipo de aplicación a desarrollar.

[Video Configuraciones MVS](#)

Funcionalidades principales del IDE

01

Editor de código

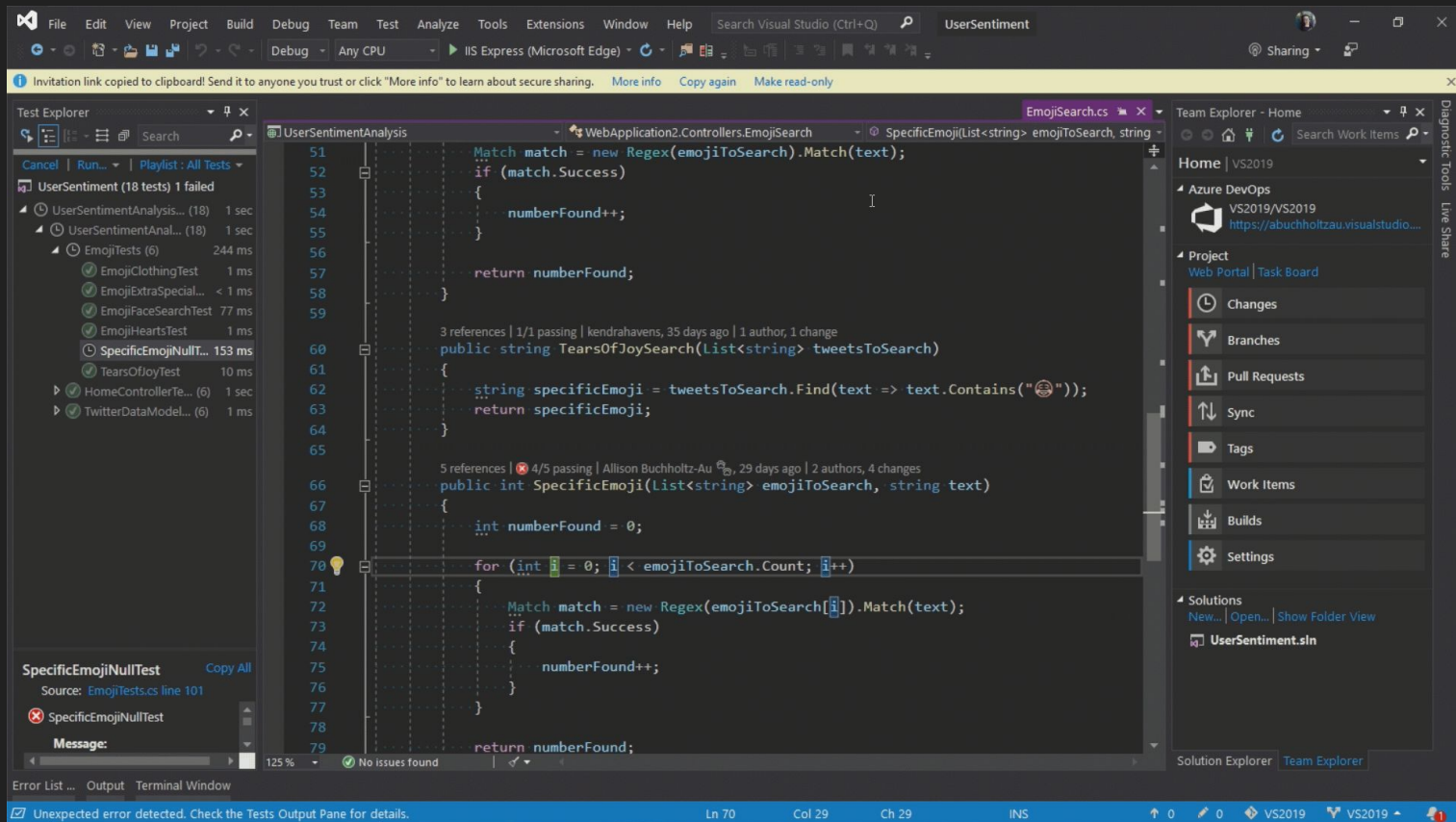
02

Debugger

03

Compilador

Editor de código



Debugger

MyProject4 (Debugging) - Microsoft Visual Studio

FILE EDIT VIEW VASSTX PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP

Process: [4800] UE4Editor.exe Thread: [4596] Main Thread Stack Frame: UFloatingPawnMovement::ApplyControlInputToVelocity(float DeltaTime)

UFloatingPawnMovement.cpp MovementComponent.cpp MovementComponent.h SceneComponent.h SceneComponent.cpp FloatingPawnMovement.h HumanFighterAIController.cpp

UFloatingPawnMovement -> UFloatingPawnMovement -> ApplyControlInputToVelocity(float DeltaTime)

```
96 {}
97
98
99
100 void UFloatingPawnMovement::ApplyControlInputToVelocity(float DeltaTime)
101 {
102     const FVector ControlAcceleration = GetPendingInputVector().GetClampedToMaxSize(1.f);
103
104     const float AnalogInputModifier = (ControlAcceleration.SizeSquared() > 0.f ? ControlAcceleration.Size() : 0.f);
105     const float MaxPawnSpeed = GetMaxSpeed() * AnalogInputModifier;
106     const bool bExceedingMaxSpeed = IsExceedingMaxSpeed(MaxPawnSpeed);
107
108     if (AnalogInputModifier > 0.f && !bExceedingMaxSpeed) { ... }
109     else
110     {
111         // Dampen velocity magnitude based on deceleration.
112         if (Velocity.SizeSquared() > 0.f)
113         {
114             const FVector OldVelocity = Velocity;
115             const float VelSize = FMath::Max(Velocity.Size() - FMath::Abs(Deceleration) * DeltaTime, 0.f);
116             Velocity = Velocity.GetSafeNormal() * VelSize;
117
118             // Don't allow braking to lower us below max speed if we started above it.
119             if (bExceedingMaxSpeed && Velocity.SizeSquared() < FMath::Square(MaxPawnSpeed))
120             {
121                 Velocity = OldVelocity.GetSafeNormal() * MaxPawnSpeed;
122             }
123         }
124
125         // Apply acceleration and clamp velocity magnitude.
126         const float NewMaxSpeed = (IsExceedingMaxSpeed(MaxPawnSpeed)) ? Velocity.Size() : MaxPawnSpeed;
127         Velocity += ControlAcceleration * FMath::Abs(Acceleration) * DeltaTime;
128         Velocity = Velocity.GetClampedToMaxSize(NewMaxSpeed);
129
130         ConsumeInputVector();
131     }
132 }
133
134
135
136
137
138
139
140
141
142
143 bool UFloatingPawnMovement::ResolvePenetrationImpl(const FVector& Adjustment, const FHitResult& Hit, const FQuat& NewRotationQuat)
144 {
145     bPositionCorrected |= Super::ResolvePenetrationImpl(Adjustment, Hit, NewRotationQuat);
146     return bPositionCorrected;
147 }
148
```

Autos

Name	Value	Type
UMovementCo	false	bool
AnalogInputM	0.000000000	const float
MaxPawnSpeed	0.000000000	const float
bExceedingMax	false	const bool
this	(0.0000000040749180, UFloating	UFloating

Call Stack

Name	Lang
UE4Editor-Engine.dll!UFloatingPawnMovement::ApplyControlInputToVelocity(float DeltaTime) Line 108	C++
UE4Editor-Engine.dll!UFloatingPawnMovement::TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction * ThisTickFunction) Line 47	C++
UE4Editor-Engine.dll!UActorComponent::ConditionalTickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction * ThisTickFunction) Line 1137	C++
UE4Editor-Engine.dll!UActorComponent::ExecuteTick(float DeltaTime, ELevelTick TickType, ENamedThread::Type CurrentThread, const TRefCountPtr<FGraphEvent> & MyComple	C++
UE4Editor-Engine.dll!FTickFunctionTask::DoTask(ENamedThread::Type CurrentThread, const TRefCountPtr<FGraphEvent> & MyCompletionGraphEvent) Line 141	C++
UE4Editor-Engine.dll!TGraphTask<FTickFunctionTask>::ExecuteTask(TArray<FBaseGraphTask * FDefaultAllocator> & NewTasks, ENamedThread::Type CurrentThread) Line 779	C++

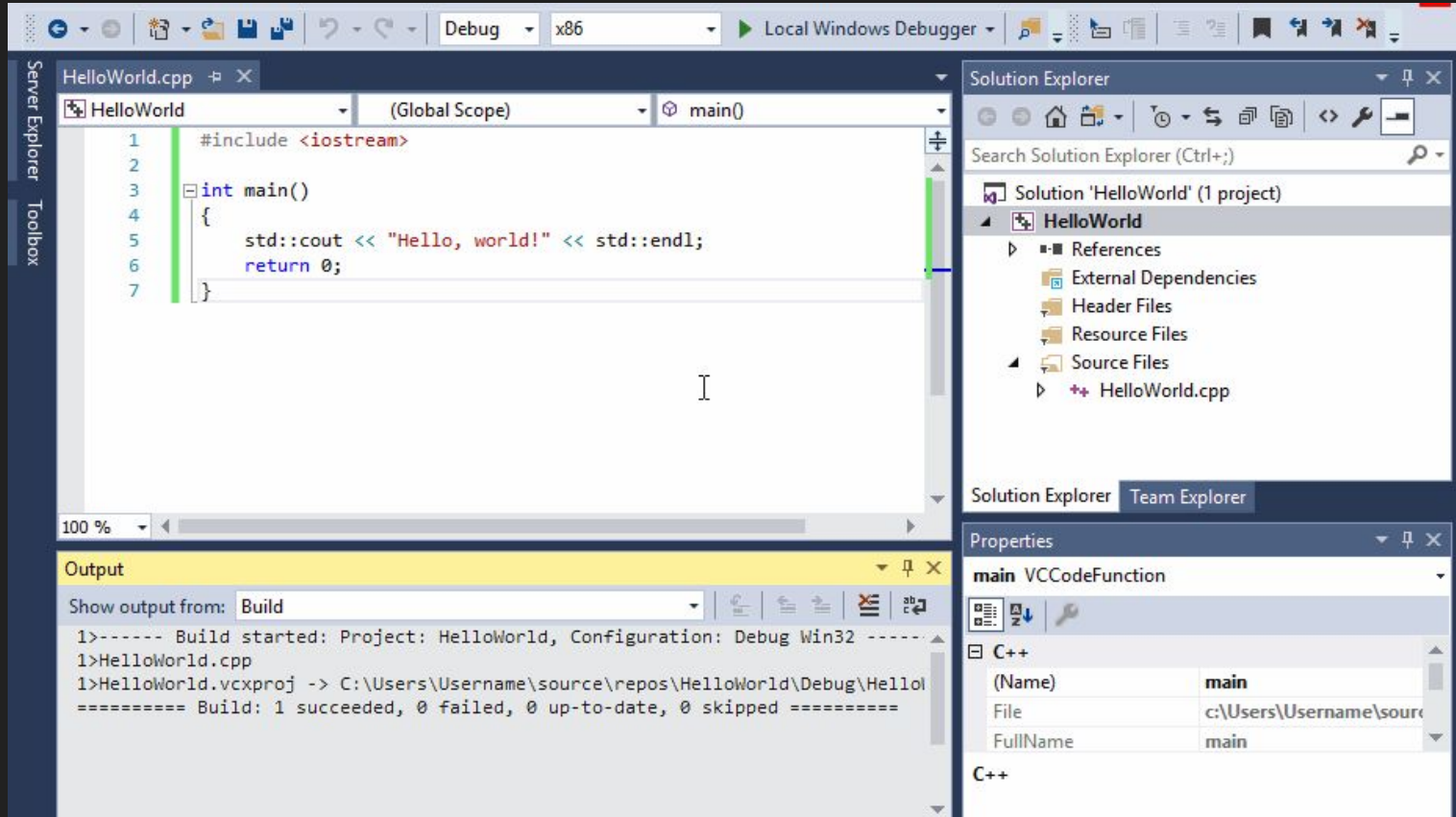
Autos Locals Watch 1

VA Find References Results Call Stack Breakpoints Command Window Immediate Window Output

Solution Explorer Team Explorer

Ln 121 Col 1 Ch 1 B45

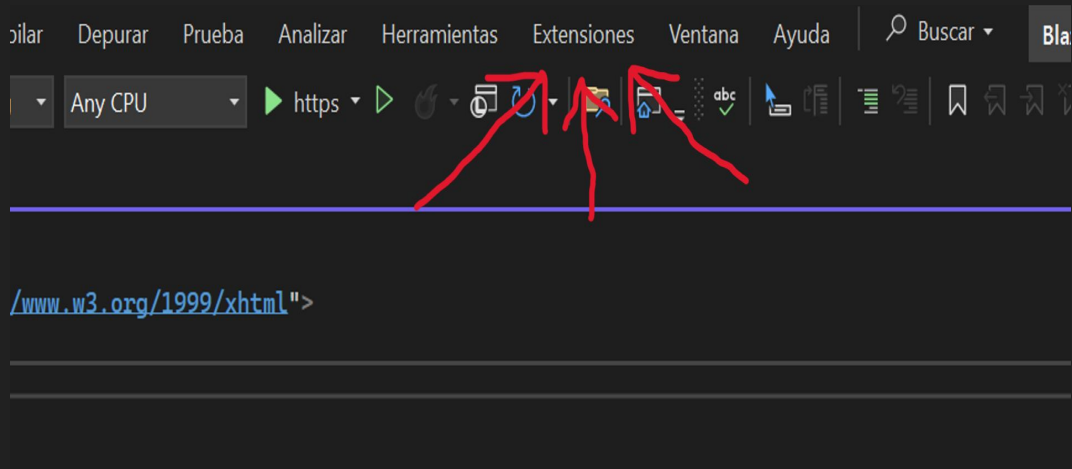
Compiler



Extensiones

Las extensiones se encuentran en la parte superior de la pantalla junto a las demás opciones que se nos permiten una vez abierto o creado un archivo en el Visual.

Tendremos acceso a diferentes extensiones según en qué ámbito nos enfoquemos a la hora de utilizar Visual.



Instalado

En línea

Visual Studio Marketplace

Actualizaciones

Roaming Extension Manager

Ordenar por: Más descargadas

Buscar (Ctrl+E)



ReSharper

The legendary .NET productivity tool: find and fix errors and code smells; navigate and refactor; run unit tests and write quality code faster.

Evaluación



Visual Assist

Productivity tool for C/C++ and C# that improves IDE features related to navigation, refactoring, code generation, and coding assistance along with specific tooling for UE4.

Evaluación

Descargar



SQLite Server Compact Toolbox

SQLite / SQL Server Compact Toolbox extension for Visual Studio. This extension adds several features to help your embedded database development efforts: Scripting of tables and data, import from SQL Server and CSV files and mu...

Evaluación



Arduino IDE for Visual Studio 2022

Fully Arduino compatible IDE. Edit, deploy and debug. Manage libraries & boards. Unique Serial Debugger for any board including full speed debugging, break/trace points with timing and function call features. Code remains compat...

Evaluación



GitHub Copilot

GitHub Copilot is an AI pair programmer that helps you write code faster and with less work. Requires Visual Studio 2022 17.5.5 or later.

Evaluación



Serene

Serenity is an ASP.NET Core / TypeScript application platform designed to simplify and shorten development of data-centric business applications with a service based architecture.



REST API Client Code Generator for VS 2022

A collection of Visual Studio custom tools for generating a strongly typed REST API Client from an Open API / Swagger

1 2 3 4 5

Creado por: Whole Tomato Software

Instala: 1470289

Categoría de precios: Evaluación

Clasificación: ★★★★★ (487 Votos)

Más información

Informar de la extensión a Microsoft

Programado para instalación:

Ninguno

Programado para actualización:

Ninguno

Programado para desinstalación:

Ninguno

Cerrar

Cambiar la configuración para las extensiones

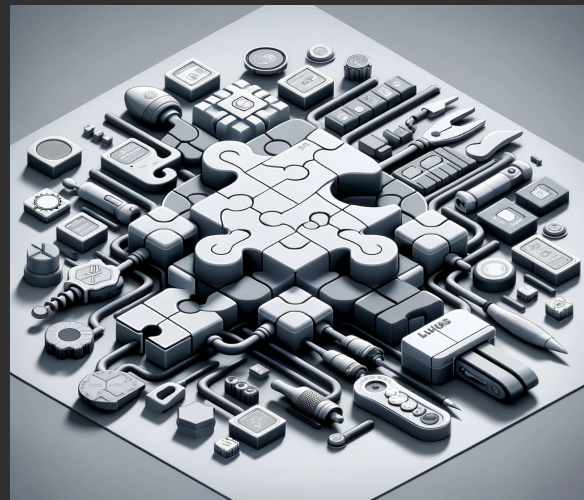
PLUGINS

En las siguientes diapositivas hablaremos de los plugins más relevantes de nuestro IDE explicaremos brevemente cada una de ellas y añadiremos link de interés muy útiles, pero antes de eso...



¿Que es un PLUGIN?

Un plugin es un software que se agrega a un programa para expandir sus funciones sin modificar su estructura principal. Permite personalizar y mejorar las capacidades del programa base.

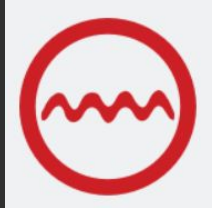




CodeMaid

CodeMaid es una extensión de código abierto para Visual Studio, diseñada para limpiar y organizar código en múltiples lenguajes como C#, C++ y JavaScript. Ofrece características como visualización en vista de árbol para archivos C# y C++, reorganización de código según convenciones y personalización de su funcionamiento. Está disponible para varias versiones de Visual Studio, incluyendo VS2019 y VS2022.

- [CodeMaid en Visual Studio Marketplace](#)
- [Página Oficial CM](#)



SonarLint

SonarLint es un plugin gratuito y de código abierto para Visual Studio que identifica y soluciona problemas de calidad y seguridad del código en tiempo real. Compatible con varios lenguajes, realiza análisis de código avanzados. Puede operar solo o integrarse con SonarQube o SonarCloud para análisis continuos en el proceso de desarrollo y CI/CD.

- [SonarLint para Visual Studio 2022 en Visual Studio Marketplace](#)
- [Documentación de SonarLint para Visual Studio](#)
- [SonarLint en SonarSource](#)

ReSharper

Proporciona análisis de código, refactorización, navegación y soporte para múltiples lenguajes de programación.

- [Guia de Instalación](#)
- [Página Oficial](#)
- [Página de descarga MICROSOFT MARKET](#)



IntelliCode

Visual Studio IntelliCode es un plugin que usa inteligencia artificial para autocompletar y predecir código en Visual Studio, basándose en patrones de proyectos en GitHub. Viene incluido en la mayoría de las versiones de Visual Studio.

- [Visual Studio Marketplace - IntelliCode](#)



Ventajas



Desventajas

- Gracias a las herramientas incluidas permite un desarrollo eficaz y menor inversión tanto en tiempo como en dinero
- A pesar de sus numerosas funcionalidades presenta un diseño bastante intuitivo para los usuarios.
- Te ofrece la posibilidad de programar en múltiples lenguajes de programación.

- Contiene muchísimas herramientas y compiladores ya instalados que lo hacen un entorno muy pesado.
- A la hora de desarrollos más complejos el IDE tiende a crashear ocasionalmente, aunque puedes recuperar el trabajo sin perderlo completamente.
- El coste de su licencia es demasiado elevado.

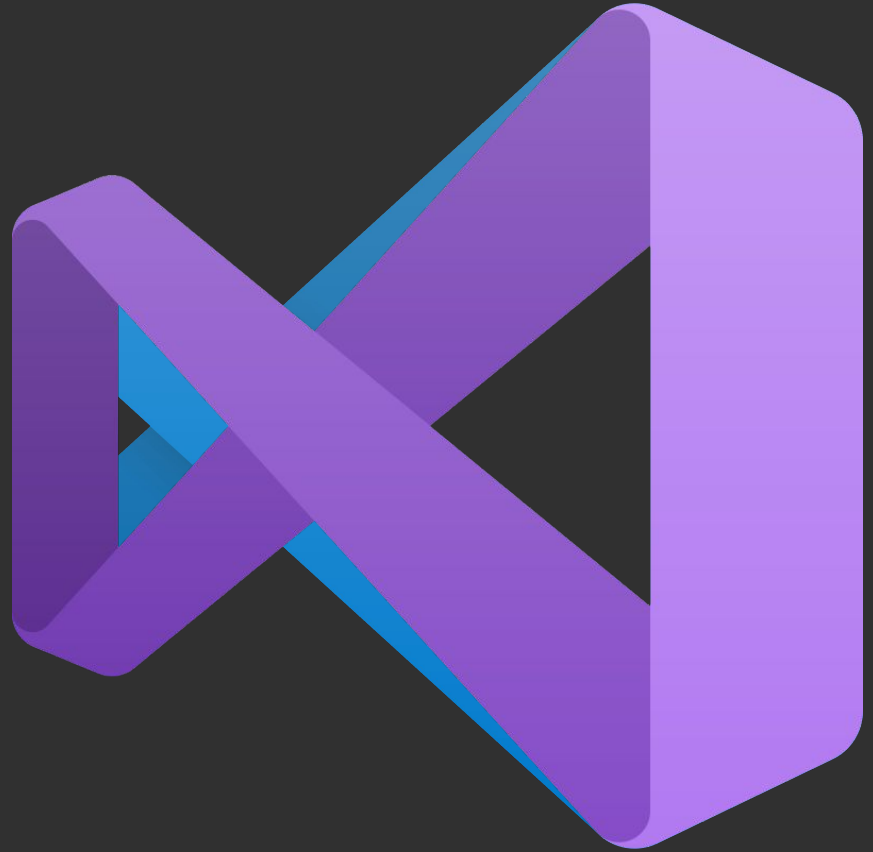
Diferencias con Visual Studio Code

Microsoft Visual Studio

- MVS es un IDE completo
- No se ejecuta en Linux
- Es un software relativamente pesado
- Las ediciones empresariales y profesionales son de pago

Visual Studio Code

- VSC es un editor de texto
- Se ejecuta en Windows, Mac y Linux
- Es un software muy ligero de base
- Es gratuito y la mayoría de sus extensiones también



FIN

Trabajo realizado por:

- David Peral
- Ibrahim Mohamed
- Mohamed Ahmed