1. Historia y contexto

Año de creación, autores o empresa desarrolladora.

Visual Studio Code fue lanzado oficialmente el 29 de abril de 2015. Fue anunciado por primera vez por Microsoft en esa fecha y, desde entonces, ha evolucionado hasta convertirse en un editor de código muy popular y ampliamente utilizado a nivel mundial.

Antes de VS Code, los desarrolladores usaban editores como Sublime Text y Atom, que eran limitados en funcionalidad y soporte para colaboración. Con el auge de la programación web y la creciente popularidad de tecnologías modernas, emergió la necesidad de un editor que pudiera abarcar estas demandas.

Desde su lanzamiento, VS Code se destacó por ser ligero, rápido y extensible, permitiendo a los usuarios personalizarlo mediante extensiones. La integración nativa con Git facilitó su adopción, y a lo largo de los años, se han agregado características como depuración avanzada, soporte para múltiples lenguajes, y herramientas de colaboración como Live Share.

Lenguajes o tecnologías a las que se orienta.

Visual Studio Code (VS Code) es un editor de código extensible, ideal para una amplia variedad de lenguajes y tecnologías, lo que lo convierte en una herramienta versátil para desarrolladores.

2. Lenguajes de Programación Soportados

VS Code ofrece soporte para varios lenguajes populares, tales como:

- **JavaScript**: Usado en el desarrollo web, tanto en el frontend como en el backend con Node.js.
- **TypeScript**: Un superset de JavaScript que agrega tipado estático, ideal para proyectos grandes.
- **Python**: Muy utilizado en desarrollo web, ciencia de datos y automatización.
- **Java**: Común en aplicaciones empresariales y desarrollo de aplicaciones Android.
- **C#:** Principalmente utilizado en aplicaciones para Windows y en desarrollo de videojuegos con Unity.
- **C/C++**: Para software de sistemas y aplicaciones de alto rendimiento.
- **PHP**: Lenguaje de scripting para desarrollo web del lado del servidor.
- **Ruby**: Conocido por su simplicidad, usado en desarrollo web con Ruby on Rails.
- **Go**: Destacado por su eficiencia y excelente soporte para programación concurrente.
- **Rust**: Lenguaje enfocado en la seguridad y rendimiento.

Tecnologías Relacionadas

Además de los lenguajes, VS Code se integra con diversas tecnologías:

- **HTML y CSS**: Fundamentales para el diseño de interfaces web en el frontend.
- **Frameworks**: Soporte para herramientas como React, Angular y Vue.js.
- **Bases de Datos**: Extensiones para trabajar con SQL y MongoDB, entre otros.
- **Contenedores**: Herramientas para gestionar Docker y Kubernetes.
- **Control de Versiones**: Integración nativa con Git para la gestión de versiones.
- **Servicios en la Nube**: Extensiones que facilitan el trabajo con plataformas como Azure, AWS y Google Cloud.

Extensiones

La capacidad de VS Code se expande mediante extensiones que permiten personalizar el entorno y añadir soporte para más lenguajes y herramientas específicas. Esto incluye linters para análisis de código, debuggers para depuración y opciones de personalización para ajustarlo a cualquier preferencia.

Evolución a lo largo del tiempo (principales versiones o hitos).

- 2015 (Noviembre): Lanzamiento inicial (v0.1.0) con edición multiplataforma básica y soporte Git.
- 2016 (Abril): Lanzamiento de la versión 1.0, marcando la estabilidad del producto e introduciendo el soporte para extensiones.
- 2018 (Febrero): La versión 1.21 introduce herramientas para trabajar con contenedores Docker.
- 2018 (Julio): La versión 1.25 introduce la funcionalidad de Live Share para colaboración en tiempo real.
- 2020 (Octubre): La versión 1.50 expande significativamente la integración con servicios en la nube.
- 2025 (Previsto): La versión 1.80 planea implementar mejoras en inteligencia artificial para autocompletado y soporte avanzado para entornos de desarrollo en la nube.

3. Instalación en una máquina virtual.

1- Sistema operativo utilizado.

El sistema operativo utilizado para instalar Visual Studio Code ha sido:

- CPU: 4 cores

- RAM: 8GB De RAM
- Disco: 40GB DISCO

2- Requisitos mínimos:

Por otro lado los requisitos mínimos que necesita visual studio code para ser instalado y utilizado son los siguientes:

- Sistemas operativos compatibles: Windows 10 o Windows 11 (versiones de 64 bits), macOS 10.15 Catalina o superior (con soporte de actualizaciones de seguridad de Apple), y distribuciones de Linux basadas en Debian/Ubuntu, Red Hat, Fedora o SUSE con arquitectura x64 (64 bits) y GLIBCXX versión 3.4.21 o superior.

- Procesador: Un procesador de 1.6 GHz o superior, con recomendación de un procesador dual-core de 2.0 GHz o superior para un mejor rendimiento.

- Memoria RAM: Al menos 1 GB de RAM para la versión de 32 bits, aunque se recomienda 2 GB de RAM para la versión de 64 bits.

Algunas fuentes indican que 1 GB es suficiente para ejecutar el editor sin problemas.

- Espacio en disco: Mínimo 1 GB de espacio libre en disco, aunque el tamaño total del instalador es inferior a 200 MB y el uso de disco es menor a 500 MB.

3. Proceso de instalación paso a paso.

1. Paso:

Para iniciar con la instalación de VSC, lo primero que deberemos hacer es ir a su página oficial, haciendo click en el siguiente enlace https://code.visualstudio.com.
![imagen](./img/Parte-1-web-visual.png)

2. Paso:

Una vez entremos en la web de VSC, lo que deberemos hacer es darle al botón de `**Download**` o también puedes usar el siguiente enlace: https://code.visualstudio.com/Download.

Una vez hagamos click en el botón de `Download` o en el enlace, nos saldra lo que vemos en la siguiente imagen:

![imagen](./img/Paso-2-Descargar-VSC.png)

Una vez abramos el enlace deberemos descargar el archivo correspondiente a nuestro sistema operativo, en esta caso como utilizamos **`Linux`** nos dirigimos a el apartado de **`Linux`** y nos descargamos el que dice **`.deb`.**

3. Paso:

Después de haber localizado nuestro archivo para **`Linux`** y le hayamos dado a descargar nos saldrá el siguiente mensaje**:**
![imagen](./img/Parte-3-Descarga-DEB.png)

Y en la parte superior nos saldrá el **`.deb`** ya descargado, una vez descargado le daremos al icono de la **`carpeta`** tal y como muestra la imagen**.**

4. Paso:

Al darle al botón nos llevará al gestor de archivos dentro de la carpeta **`Descargas`** donde se encuentra el **`.deb`.**
![imagen](./img/Parte-4-Localizar-Archivo-DEB.png)

4. Paso:

Después de localizar el localizar nuestro **`.deb`,** deberemos abrir una terminal y para ello tenemos dos formas:

- Primera forma: Podemos dirigirnos a una CMD usando el comando **`CONTROL + ALT +T`,** y una vez dentro de ella ejecutar el comando **`cd Descargas/`** o en algunos casos **`cd Download`.**

- **Segunda forma:** Desde el mismo explorador de archivos usando click derecho y darle a la opción a la de **`Abrir en una Terminal`** y ya estaremos dentro de la ruta necesaria**.**

Aquí os dejamos una foto de como hacer la segunda forma:
![imagen](./img/Parte-4-Abrir-Una-Terminal.png)

5. Paso:

Una vez realizado todos estos pasos y estar dentro de una **`Terminal`** y dentro de la ruta **`Descargas`** podremos ejecutar el siguiente comando para poder instalar VSC **`sudo apt install ./<file>.deb`** tal y como se muestra en la imagen**.**

![imagen](./img/Paso-5-Instalación.png)

Una vez le damos al **`enter`** en la terminal es posible que nos muestre este mensaje, donde nos pida descargar ciertos recursos externos, si esto ocurre simplemente vuelva a darle enter.

6. Paso:

Una vez terminemos la instalación podemos ir al menú de inicio y buscar **`Visual Studio Code`** y este nos aparecerá instalado tal y como aparece en la imagen:
![imagen](./img/Paso-6-Instalacion-completada.png)

4. Partes y uso del IDE: documentar cada una de las partes en el README GRECIA:

- Editor de código: principales características.

    1. Ligero y Rápido: VS Code es un editor de código muy ágil y eficiente, ideal para proyectos de diferentes tamaños.
    2. Soporte Multilenguaje: Soporta una gran variedad de lenguajes de programación como JavaScript, Python, C++, Java, HTML, CSS, y muchos más, mediante extensiones.
    3. Extensibilidad: Cuenta con un marketplace de extensiones que permite agregar funciones como depuración, temas, fragmentos de código, control de versiones, entre otras.
    4. Depuración Integrada: Incluye herramientas de depuración que facilitan la identificación y corrección de errores en el código.
    5. Control de Versiones Integrado: Integración nativa con Git, permitiendo gestionar repositorios, realizar commits, ramas y otras operaciones desde el mismo editor.
    6. Terminal Integrado: Incluye una terminal de línea de comandos dentro del editor, lo que facilita el trabajo sin cambiar de ventana.
    7. IntelliSense: Función de autocompletado inteligente que ofrece sugerencias de código, documentación y parámetros en tiempo real.
    8. Refactorización y Navegación: Herramientas para refactorizar código fácilmente, buscar y navegar entre archivos y símbolos rápidamente.
    9. Temas y Personalización: Permite cambiar la apariencia y comportamiento mediante temas, iconos y configuraciones personalizadas.
    10. Soporte para Docker y Contenedores: Facilita el trabajo con entornos de contenedores y desarrollo en la nube.

- Compilador / Intérprete: cómo se ejecutan y prueban los programas.
  En Visual Studio Code, la ejecución y prueba de programas depende del lenguaje de programación que estés utilizando y del entorno configurado. VS Code en sí no incluye un compilador o intérprete integrado, pero se puede configurar para usar las herramientas específicas del lenguaje.
    1. Configuración del entorno

- Instalar el compilador o intérprete: Debes instalar en tu sistema las herramientas necesarias para tu lenguaje (por ejemplo, GCC para C/C++, Python, Node.js para JavaScript, etc.).
- Configurar VS Code: Muchas veces necesitas agregar tareas o configuraciones específicas en tasks.json o launch.json para compilar y ejecutar programas.

2. Compilar / Interpretar y Ejecutar

- Para lenguajes compilados (por ejemplo, C, C++, Java):
- Compilación: Ejecutas un comando de compilación (como gcc archivo.c -o archivo) en la terminal integrada de VS Code.
- Ejecución: Luego, corres el programa compilado desde la terminal (./archivo en Linux/Mac o archivo.exe en Windows).
- Para lenguajes interpretados (como Python, Ruby, JavaScript):
- Interpretación: Ejecutas directamente en la terminal integrada (python archivo.py, node archivo.js).

3. Automatización con tareas y depuración

- Puedes crear tareas automatizadas en tasks.json para compilar y ejecutar tu código fácilmente.
- También, puedes usar las configuraciones de depuración (launch.json) para correr y depurar programas paso a paso.

4. Extensiones específicas

- Muchas extensiones (como la de Python, Java, C++) proporcionan botones y comandos integrados para compilar, ejecutar y depurar, simplificando mucho el proceso.

    - Depurador: uso básico de breakpoints, inspección de variables, etc.

1. Configurar la depuración
    - Abre tu proyecto y ve a la pestaña de Ejecutar y depurar (el ícono de un triángulo con un insecto en la barra lateral izquierda o presiona Ctrl+Shift+D).
    - Si es la primera vez, necesitas crear una configuración de depuración:
    - Haz clic en crear un archivo launch.json.
    - Selecciona el entorno adecuado (por ejemplo, Node.js, Python, C++, etc.).
    - VS Code generará una configuración básica para tu lenguaje.
2. Establecer breakpoints
    - Haz clic en el margen izquierdo junto a la línea de código donde quieres detener la ejecución (aparecerá un punto rojo).
    - También puedes presionar F9 con el cursor en la línea deseada.
    - Los breakpoints indican a VS Code que debe pausar la ejecución en esas líneas.
3. Iniciar la depuración
    - Presiona el botón de Iniciar depuración (el triángulo verde) o presiona F5.

- La ejecución se detendrá en el primer breakpoint alcanzado.

4. Inspección de variables y estado
   - Cuando la ejecución está pausada:
   - Ventana de Variables: Muestra las variables en el alcance actual.
   - Hover: Pasa el ratón sobre una variable para ver su valor.
   - Panel de Inspección: Puedes agregar expresiones manualmente para monitorear su valor.
   - Consola: También puedes evaluar expresiones en la consola de depuración.
5. Controles de ejecución
   - Continuar (F5): Continúa la ejecución hasta el siguiente breakpoint.
   - Paso a paso:
   - Paso sobre (F10): Ejecuta la siguiente línea de código.
   - Paso en (F11): Entra en funciones o métodos.
   - Paso fuera (Shift+F11): Sale de la función actual.
   - Detener (Shift+F5): Termina la sesión de depuración.

JULIAN:

- **Control de versiones: integración con Git o sistemas similares.**
  - **Integración:** VS Code tiene soporte nativo para Git, permitiendo realizar commits, push y pull desde la interfaz.
  - **Visualización de Cambios:** La pestaña de control de versiones muestra archivos modificados, nuevas adiciones y permite ver diferencias.
  - **Resolución de Conflictos:** Si se producen conflictos, VS Code ofrece herramientas visuales para gestionarlos.
- **Refactorización: ejemplos prácticos de uso.**
  - **Renombrar Variables:**Cambiar el nombre de una variable en todo el proyecto sin errores.
  - **Extraer Métodos:** Seleccionar un bloque de código y convertirlo en un método separado para mejorar la legibilidad.
  - **Mover Clases o Métodos:** Trasladar elementos a diferentes archivos o paquetes organizativos.
- **Plugins o extensiones: principales complementos disponibles.**
  - **Prettier:** Formatea automáticamente tu código al guardar. Es compatible con múltiples lenguajes como JSON, JavaScript, CSS y Markdown. Con Prettier, tus archivos mantienen un estilo uniforme sin discusiones innecesarias de formato.
  - **ESLint:** Detecta errores de sintaxis, malas prácticas y problemas de estilo en tiempo real para proyectos JavaScript y TypeScript. Gracias a esta extensión, podés escribir código más robusto antes de siquiera ejecutarlo.

- **Live Server:** Capacidad para ver cambios en tiempo real en el navegador.
- **GitLens:** Mejora la visualización de la historia de cambios en Git y detalles de cada commit. Además con GitLens, el historial de tu proyecto se vuelve visual y útil. Como anteriormente dicho visualizá cambios línea por línea, accedé a detalles de commits, ramas y anotaciones “blame” sin salir del editor.

4. Ejemplo práctico

- Breve demostración del desarrollo de un pequeño proyecto o script (por ejemplo, “Hola mundo” o una función simple).

- **Java(“Hola Mundo”):**

```
public class HolaMundo {
    public static void main(String[] args) {
        System.out.println("Hola, Mundo");
    }
}
```

- **Python(“Hola Mundo”):**

```
print("Hola, Mundo")
```

- **C++(“Hola Mundo”):**

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hola, Mundo" << endl;
    return 0;
}
```

- **C#(“Hola Mundo”): using System;**

```
class Program {
```

```
    static void Main() {
        Console.WriteLine("Hola, Mundo");
    }
}
```

5. Valoración personal o grupal

VS Code es excelente por ser rápido y muy personalizable gracias a sus extensiones. Su contra es que necesita configuración manual para igualar la potencia de los IDEs tradicionales.

-Ventajas y Desventajas

Ventajas:

1. **Ligero y Rápido:** Arranca casi al instante y consume pocos recursos del sistema.
2. **Extensibilidad:** Su Marketplace de extensiones permite adaptarlo a casi cualquier lenguaje o framework (Python, Java, C++, etc.).
3. **Integración con Git:** El control de versiones está integrado de forma nativa y muy intuitiva.

Desventajas:

1. **Configuración Inicial:** Para tener una experiencia completa de IDE (como depuración avanzada o IntelliSense perfecto), necesitas instalar y configurar varias extensiones.
2. **Menos Funcionalidad Nativa:** Comparado con IDEs completos como Visual Studio o IntelliJ, algunas herramientas avanzadas (como refactorización compleja) requieren extensiones adicionales.

-Contextos donde se recomienda usar Visual Studio Code

Se recomienda usar Visual Studio Code cuando necesitas un editor rápido y versátil que se adapte a múltiples tareas de desarrollo.

1. **Desarrollo Web Front-end y Back-end:** Es el estándar para trabajar con JavaScript, TypeScript, React, Node.js y Python, gracias a su excelente soporte nativo y a las extensiones específicas para estos ecosistemas.
2. **Trabajo Multi-Lenguaje:** Si cambias frecuentemente entre lenguajes (por ejemplo, haces un poco de Python, luego haces algo de Bash y después editas archivos de configuración JSON/YAML), VS Code gestiona todos estos contextos sin ralentizarse.

3. **Trabajo Remoto o en Máquinas con Recursos Limitados:** Su bajo consumo de memoria y CPU lo hace ideal para usar en máquinas virtuales, entornos de **cloud computing** o portátiles menos potentes.
4. **Scripts y Automatización:** Es perfecto para escribir y ejecutar **scripts** rápidos, tareas de automatización, ya que no tienes que esperar a que se cargue un IDE completo.