

Trabajo EED sobre VS Code, NetBeans y Eclipse

1. Breve explicación de requisitos e instalación de cada uno
2. Similitudes y diferencias entre los 3
3. Funcionamiento de los 3, ¿cuál es más fácil y difícil de usar?

1.A Requisitos de instalación

- Todos requieren de tener suficiente CPU RAM y espacio (los cuales son relativamente similares para los 3 entornos a la hora de desarrollar las mismas actividades, por ejemplo, para programar en java todos necesitan un JDK "Java Development Kit"). Aun así, es cierto que VS Code tiende a ser algo más ligero que los otros 2 a cambio de tener menos funcionalidades para el desarrollo.
- Es importante tener en cuenta el tamaño de los proyectos en los que vas a trabajar a la hora de considerar los requisitos (a proyectos más grandes más requerimiento de espacio RAM y CPU)
- Como VS Code funciona más como una "versión Super-Mega-Turbo" de Notepad++, como a mí me gusta decir, este sirve para cualquier tipo de lenguaje, mientras que los otros 2 están más centrados en torno al desarrollo en java y los lenguajes y herramientas complementarios que esto requiera.

1.B Método de instalación

Visual Code *

1. Descargar VS Code:

- Ve al sitio web oficial de Visual Studio Code: [Visual Studio Code](https://code.visualstudio.com/).
- Descarga la opción que corresponda a tu sistema operativo (Windows, macOS o Linux).

2. Instalar VS Code:

- Para Windows:
 - Ejecuta el instalador descargado (.exe).
 - Sigue las instrucciones en pantalla para completar la instalación.
- Para macOS:
 - Abre el archivo .zip descargado.
 - Arrastra la aplicación de VS Code a la carpeta de Aplicaciones.
- Para Linux:
 - Sigue las instrucciones específicas de tu distribución de Linux, disponibles en el sitio web de VS Code.

3. Abrir VS Code:

- Una vez que la instalación esté completa, inicia VS Code abriéndolo desde la carpeta de instalación o utilizando el acceso directo creado durante la instalación.

4. Extensiones (Opcional):

- VS Code admite extensiones para mejorar su funcionalidad. Explora e instala extensiones según tus necesidades.
 - Haz clic en el icono de Extensiones en la barra lateral (o utiliza el atajo **Ctrl+Shift+X**).
 - Busca extensiones e instala aquellas que desees.

5. Actualizar Configuraciones (Opcional):

- Personaliza las configuraciones de VS Code accediendo al menú de configuración (Archivo > Preferencias > Configuración o con **Ctrl+,**).
- Modifica las configuraciones según tus preferencias.

6. Explora y Aprende:

- Familiarízate con las características de VS Code, los atajos de teclado y la documentación.
- Utiliza la terminal integrada, el control de versiones y otras herramientas.
- Vs Code tiene buenos tutoriales para facilitarte el comienzo de su uso, aunque la mayoría están en inglés así que es recomendable tener un buen nivel de este para que la experiencia de aprendizaje sea más fluida y cómoda.

* Lo sé, esto es casi calcado de ChatGPT 3.5, pero como estaba mayor mente correcto (salvo algunos cambios que le he hecho) he considerado correcto usarlo.

Eclipse y NetBeans

1. Descargar Eclipse/NetBeans desde la página web oficial (hay varios paquetes a escoger en función del propósito que le quieras dar al IDE)
2. Descargar e instalar el JDK, el cual puede ir incluido en el paquete de instalación, se puede instalar durante la instalación del IDE si el asistente te da la opción o se puede descargar manualmente de una distribuidora (habitualmente OpenJDK u Oracle). Al instalarlo se requiere habitualmente de configurar el JDK para poder hacer un correcto uso de este.
3. Instalar el IDE a través del .exe descargado (si se ha descargado un .zip hay que descomprimirlo primero) y seguir las instrucciones seleccionando las configuraciones que uno desea para el IDE (paquetes de ayuda a instalar, PATH del JDK y/o versión de este, etc).
4. Abrir el IDE por primera vez para iniciar el proceso de configuración y las guías iniciales de uso de la herramienta (esta configuración es diferente de la de instalación porque esta se refiere más a las preferencias/ajustes del IDE y la otra más a las herramientas del IDE)

2.A Similitudes

- Son todos multilenguaje (a través de la instalación de extensiones/paquetes/plugins/etc).
- Todos tienen una estructura de GUI (Graphical User Interface) similar en cuanto a la edición y prueba de código y en cuanto al manejo de proyectos.
- Todos permiten una alta personalización estética y funcional comparada con la de otros programas. No por nada son programas pensados para ser usados por programadores cualificados (aunque ese no sea su único público).
- Los requerimientos son similares en los 3 si se quiere desarrollar proyectos en Java.

2.B Diferencias

- Eclipse y NetBeans están más orientados al desarrollo en Java mientras que VS Code es más minimalista (no se puede desarrollar bien en ningún lenguaje sin descargar plugins).
- Por su naturaleza generalista y minimalista, VS Code es como un lienzo en blanco, lleno de posibilidades y opciones que lo hacen extremadamente personalizable comparado con los otros 2, aunque todos permiten el uso de plugins para modificar la experiencia de uso (modo oscuro) y el funcionamiento (programar en C++ en vez de Java).
- La instalación de Eclipse y Apache NetBeans es relativamente lineal y sencilla hasta el momento en el que te puedes poner a programar, pero sus entornos pueden ser obtusos y confusos si no conoces cómo funciona un IDE, mientras que VS Code requiere de más pasos para terminar de instalarse y configurarse, pero tiene muchas guías y tutoriales a parte de una interfaz que, aunque similar a la de los otros 2, es más intuitiva y clara.

3. Funcionamiento

El funcionamiento de los 3 es bastante similar ya que todos son IDEs, aunque es cierto que existen diferencias en su uso. Apache NetBeans es usado junto con VS Code para enseñar programación a alumnos, mientras que eclipse es más usado en entornos de empresa. Además, Eclipse y NB están orientados más al desarrollo en Java mientras que VS Code es más una herramienta multiusos.

De todas formas, la realidad es que no hay ninguno mejor o peor, sino que sencillamente depende del proyecto y la familiaridad del equipo de desarrollo con el entorno y sus herramientas. Es similar a lo que sucede con los lenguajes de programación, en realidad se puede hacer cualquier programa en cualquier lenguaje (aunque es cierto que cada uno tiene sus especialidades), lo cual implica que muchas veces se escoge con lo que uno esté más cómodo a menos que las ganancias de cambiar sean claramente significativas.

De esto se puede sacar la correcta conclusión de que no existe un entorno más fácil que otro de usar como tal. Todo depende más de la documentación, el soporte del creador y de la comunidad al IDE y de gustos personales. En el caso que nos compete, los 3 entornos tienen apoyo de sobra para que uno pueda aprender a hacerlos funcionar, así que el ranking de dificultad de uso se vuelve más un tema personal que otra cosa.