



---

# COMPARACIÓN DE ENTORNOS DE DESARROLLO

---

Entornos de Desarrollo – A2.1



Marco Valiente  
DAW 1

## Contenido

Introducción .....	2
Desarrollo .....	3
1. Características de Eclipse .....	3
2. Características de Visual Studio Code .....	3
3. Características de IntelliJ .....	4
4. Semejanzas entre ellos.....	5
5. Diferencias entre ellos.....	5
Conclusiones .....	6
Bibliografía .....	7

## Introducción

En este documento, vamos a comentar las diferencias y similitudes dentro de los diferentes entornos de desarrollo, como pueden ser Eclipse, IntelliJ y Visual Studio Code. Para realizar esta evaluación, vamos a buscar individualmente acerca de cada propio entorno de desarrollo, así como listando tanto su uso como sus características y utilidades actuales.

Todas las aplicaciones que se van a describir aquí califican como IDE, que significa Integrated Development Environment, que permiten la edición, prueba, y corrección de código en la misma aplicación, sin necesidad de herramientas externas.

Una vez que sea completado, quedará, como último apartado y cierre de la actividad, una barra de conclusiones donde se evaluarán unas conclusiones finales después de esta actividad.

# Desarrollo

## 1. Características de Eclipse

Eclipse es un IDE desarrollado por una compañía de non-profit de la unión europea llamada Eclipse Foundation. Su código es desarrollado y distribuido en código abierto.

Este posee las siguientes características:

- **Soporte para las últimas versiones de Java.** Ofrece hasta la nueva versión de Java 25.
- **Herramientas para programación en Java.** Tiene soporte a JUnit 6.0.1, una API con una asistencia en programación en Java y diseñada para gestión de proyectos (workspaces).
- **Extensibilidad.** Permite la adición de Plugins a través de un Marketplace.
- **Adaptación a plataformas.** La aplicación se adapta a los diferentes monitores y está optimizada para que funcione más rápido, con mejores interfaces y rendering.
- **Código Abierto y Gratuito.** El código se publica con una licencia pública de uso.
- **Soporte para extensiones.** Los plugins tienen soporte para poder ser incorporados con las herramientas de Java, como JUnit 6.
- **Contribución Libre.** A pesar de que se desarrolla por Eclipse Foundation, al ser de código abierto, se permite a los propios desarrolladores la publicación de contribuciones al mismo código de Eclipse IDE por la propia comunidad.

Como podemos ver, pues, por todas estas características, se trata de un IDE que se centra más en la comunidad a la hora de realizar cambios o publicaciones, especialmente centradas en Java, con una orientación dirigida al desarrollo de forma libre.

## 2. Características de Visual Studio Code

Visual Studio Code, también llamada VSCode es una plataforma desarrollada por Microsoft y que también es de código abierto, aunque su enfoque más reciente se ha centrado en la incorporación de la inteligencia artificial dentro del propio código, como para predicciones de las líneas que el usuario deseará escribir.

Este posee las siguientes características:

- **Soporte de modelos de IA.** Por los propios miembros de VSCode, ellos ofrecen diferentes opciones a la hora de usar el programa, como pueden ser IAs que tiene acceso al código que estés escribiendo, para poder diseñar respuestas que requieran menos contexto para la respuesta que estés buscando.
- **Código en cualquier lenguaje.** VSCode permite a sus usuarios programar tanto en JavaScript, Python, C, C++, TypeScript, HTML, Java, etc.

- **Extensibilidad.** Posee un Marketplace con diferentes accesos a plugins creados tanto por la comunidad como por los desarrolladores de por sí.
- **Personalizable.** Permite la creación y personalización de los entornos donde se trabaja, guardando cualesquiera modificaciones dentro de perfiles los cuales pueden cambiarse con solo presionar un botón.
- **Acceso Gratuito.** El acceso a VSCode desde tanto un enfoque personal como un enfoque de empresas a la hora de realizar programación es el mismo, acceso de descarga gratuito por el cual cualquier persona puede acceder a este.
- **Código a través de la Web.** Al pertenecer a Microsoft, misma compañía que desarrolla GitHub actualmente, permite la interconexión con Git para permitir escribir en código incluso desde la propia web, desde <https://vscode.dev/?vscode-lang=es-es>.

El entorno de desarrollo de VSCode es, por tanto, uno que se centra en su flexibilidad, aunque a nivel de desarrollo, este sea inferior en algunos lenguajes específicos, pero compensa esto gracias a su capacidad de ser personalizable de forma abierta, de ser de acceso gratuito, y de ofrecer una gran cantidad de libertad a la hora de programar en una amplia cantidad de lenguajes de programación.

### 3. Características de IntelliJ

IntelliJ es un IDE proveniente de JetBrains, una compañía externa localizada en Chequia creada por 3 personas de nacionalidad rusa y actualmente localizada en Países Bajos, la cual se centra en ser una herramienta de desarrollo inteligente para los desarrolladores. En cuanto se refiere a desarrollo serio en Java, este programa se considera el mejor existente del mercado, y ya que este programa se ha desarrollado con Java en mente.

Este posee las siguientes características:

- **Exclusividad de Java y Kotlin.** Este IDE está diseñado para prioritariamente trabajar en estos lenguajes de programación.
- **Código Rápido.** Encendido por defecto, el sistema de IntelliJ ofrece una compleción de código dinámica centrada en el contexto por el cual se estén operando. También tiene inyecciones y plantillas que te ofrecen para autocompletar estructuras específicas de código.
- **Navegación Específica.** Permite encontrar y buscar el uso de tanto tus propios códigos como los frameworks que provengan de librerías. También describe mientras se esté escribiendo la documentación para poder leer los detalles específicos de esa framework específica.
- **Inteligencia Artificial.** IntelliJ ofrece inteligencia artificial para asistir con la programación dentro del mismo, así como sugerencias de código o incluso de compleción de código, aunque esa viene desactivada por defecto hasta que se acepten los términos y condiciones de uso de la misma.

- **Integración dentro del IDE.** IntelliJ posee tanto un debugger de código, tanto como un profiler para poder detectar cómo de rápido o eficiente el código siendo desarrollado es, así como soporte de una terminal integrada directamente dentro del propio programa, junto con soporte a Git y similares.
- **Facilidad de uso.** Es capaz de ejecutar y debugear códigos tanto en tu propio ordenador como con la creación de máquinas virtuales pequeñas que se usen para ejecutar el código como medida de seguridad. El código ejecutado se considera altamente eficiente, especialmente en consideración con ejecuciones grandes.
- **Entorno personalizable.** Te permite tanto adaptarte a entornos, como para cambiar la apariencia del propio IDE. Posee, así mismo, un Marketplace con plugins.

Una cosa a destacar es, que, mientras existe código abierto y una versión gratuita ejecutable, también existen límites dentro de esa versión, ya que existen licencias que te desbloquean mayores funcionalidades potentes para el uso en programación, un detalle considerable. Este programa se considera el número 1 líder en programación en Java. Las estructuras de sus pantallas y opciones a veces son consideradas mejorables.

#### 4. Semejanzas entre ellos

Estos tres IDEs tienen en común que son de código abierto (al menos en las versiones gratis) y ofrecen para programar los entornos de código junto con comunidades de plugins para las diferentes necesidades, construidas todas ellas gracias a comunidades que desarrollan dichos plugins. Todos permiten cierto nivel de personalización, aunque algunas ofrecen de mejor manera que otras, y permiten la edición, así como la ejecución de código del mismo. Todos ellos tienen auto-completación de código, aunque en distintos niveles, y todos ellos son multiplataformas que se pueden conectar con Git.

#### 5. Diferencias entre ellos

Lo más evidente que se puede decir para diferenciarlos, es que IntelliJ cuesta dinero para tener habilitadas todas sus funciones, mientras que tanto Eclipse como VSCode son gratuitas de acceso libre.

VSCode es orientada más a mayor libertad para la edición de cualquier tipo de archivos y de lenguajes de programación, reluciendo en archivos pequeños.

Eclipse es una ejecución de Java más extensible, y el cual su principal foco es Java, pero resulta más pesado y tarda un mayor tiempo a la hora de realizar ejecuciones de código. Se centra en ser una opción más fuerte en los aspectos de Java pero sin especificarse en uno en particular.

IntelliJ se centra en tener un buen análisis del código, junto con la mejor auto-completación posible del mismo, además de tener un gran soporte para proyectos grandes que tiene mínima tolerancia a errores.

## Conclusiones

Podemos ver cómo estos diferentes programas tienen diferentes niveles de profundidad y de desarrollo, y aunque algo más en detalle como la propia eficiencia base no se puede cambiar, al tener todos ellos la capacidad de usar plugins y de ser personalizados para tareas específicas, les otorgan siempre al final la elección al usuario de cómo quiere que funcione el programa, si está haciendo suficiente, si prefieren que el programa sea capaz de hacer más, y les den, independientemente del IDE que sea, ese nivel de control.

Por eso, aunque haya algunos IDEs que sean mejores que otros, como, por ejemplo, como se ve que una gran parte de usuarios tanto de principiantes como de expertos, aprecian más al IDE de IntelliJ, la opción de usar un programa con VSCode con los suficientes plugins al final puede ser capaz de lograr un objetivo similar.

## Bibliografía

- <https://eclipseide.org/>
- <https://docs.junit.org/current/api/>
- <https://code.visualstudio.com/>
- <https://marketplace.visualstudio.com/vscode>
- <https://www.jetbrains.com/idea/features/>
- <https://dev.to/nisachampagne/intellij-vs-eclipse-vs-vscode-1g09>