

## 7 plataformas diferentes para desarrollar Android Apps

Por [Ivan Lasso](#) -

abril 27, 2017



Elegir el IDE adecuado para desarrollar una aplicación para Android **puede marcar la diferencia** entre obtener un producto de calidad o uno inferior. Y no solo eso, sino que un desarrollador puede encontrar que su vida se hace más sencilla dependiendo de la elección que tome, teniendo una experiencia frustrante o fluida y ágil de acuerdo a la opción por la que se decante.

En [Android Authority](#) elaboraron una lista con **diversos IDE y métodos** para desarrollar aplicaciones para Android. Hemos extraído algunos de ellos y los traemos aquí para ayudarles no solamente a elegir la mejor opción para ustedes, sino también para ayudarles a conocer parte de lo que está disponible en el mercado. Quién sabe, tal vez entre estas opciones esté alguna que desconocían y que se adecue al proyecto que tengan entre manos.

### [Android Studio](#)

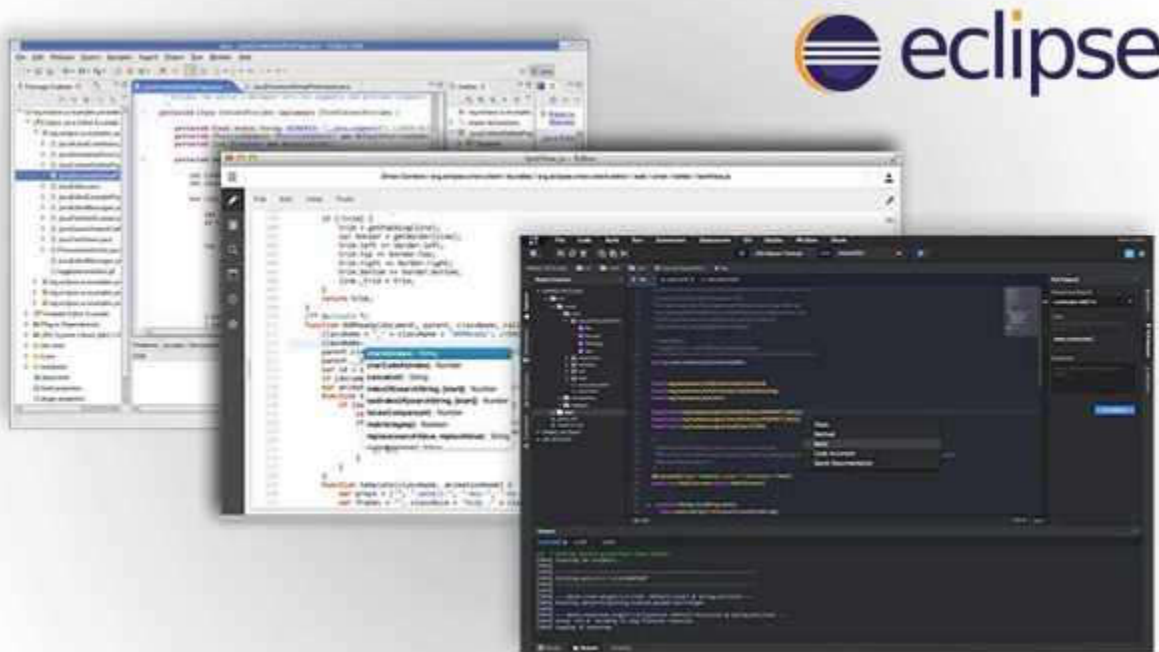


Empezamos por el **IDE oficial creado por Google** y al que hace referencia toda la documentación oficial existente. Es por ello que es el entorno que con más facilidad permite seguir las directrices de diseño de Google y encontrar nuevas librerías para nuestros proyectos. Y aunque resulta un poquito complicado de configurar, lo cierto es que va mejorando con rapidez y ya es mejor de lo que era antes, especialmente cuando salió.

En este IDE, **se desarrolla en Java utilizando referencias al Android SDK** (Software Development Kit), lo cual facilita acceder a elementos de la interfaz de usuario para visualizarlas en una vista de diseño que es muy útil.

Android Studio es una gran opción de cara a **desarrollar servicios y herramientas de productividad**, aunque también permite el desarrollo de juegos sencillos. Si se quiere hacer algo que necesite el uso de una física más avanzada, se puede recurrir a librerías como LibGDX para que no haya que hacer todo desde cero, pero si lo que se quiere hacer es más complejo, será necesario conocer Android NDK (Native Development Kit) que permite usar C++ para un mayor uso de la CPU, algo que no es tan poderoso como otras opciones.

## [Eclipse](#)



Al principio, Eclipse era el IDE recomendado para el desarrollo de aplicaciones. De hecho, fue la opción oficial que se planteaba desde Google hasta la aparición de Android Studio. Desde el lanzamiento de este último, se suele recomendar que los desarrolladores que utilicen Eclipse **migren hacia él**.

La configuración de Eclipse y el flujo de trabajo **son muy parecidos** a los de Android Studio. Pero el hecho de que no sea un entorno de desarrollo específico para la plataforma de Mountain View, sino que está pensado para desarrollar en varias plataformas e idiomas, hace que la experiencia en general sea más lenta y que haya cierta tendencia a que se produzcan errores.

## [Xamarin](#)



Si lo que se busca es desarrollar una aplicación multiplataforma, que permita, por ejemplo, partir del mismo código para obtener un producto que **también salga para iOS y Windows**, Xamarin es una buena opción. Es un entorno creado por Microsoft, gratuito y que viene incluido en Visual Studio.

**En Xamarin se desarrolla en C#** y, aunque utilizarlo significa alejarse de la experiencia de desarrollo «puro» para Android, es una alternativa a considerar para aquellos que Java no les guste o no sea su fuerte. Entre algunas de las cosas interesantes con que cuenta están las pruebas automatizadas a través de múltiples dispositivos reales conectados a la nube y el soporte de Monogame, un framework multiplataforma para juegos basados en el framework XNA de Microsoft, una gran herramienta para hacer juegos 2D y 3D.

## [AIDE](#)



AIDE, que significa Android IDE, se diferencia del resto de entornos aquí reseñados en que es **se ejecuta directamente en Android**, de manera que es posible desarrollar una aplicación desde el mismo dispositivo móvil (como el celular o la tablet) y hacer pruebas allí sin necesidad de un emulador u otro dispositivo pensado únicamente para ello. Por otra parte, su funcionamiento es bastante similar al de Android Studio y Eclipse.

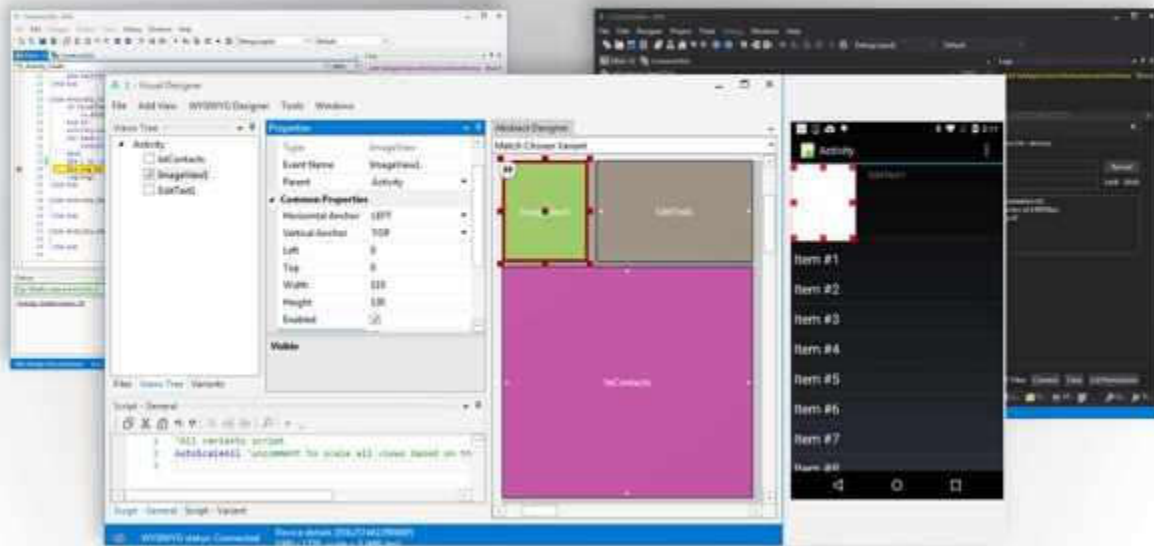
Aunque cuenta con una **serie de tutoriales bastante buenos**, tal vez no es la mejor opción al menos de cara a proyectos más o menos grandes. Al fin y al cabo, trabajar en una pantalla pequeña, incluso en la de una tablet, es como poco incómodo. Además, carece de algunas funcionalidades y no hay una forma real de hacer pruebas en varios dispositivos. Aún así, es una buena opción para aprender los fundamentos del desarrollo de aplicaciones, sobre todo por la facilidad de probar los resultados de cada paso que des, aunque a las lecciones más avanzadas solo se puede acceder por medio de suscripción.

## [Python](#)



Este popular lenguaje de programación puede ser también utilizado para desarrollar aplicaciones de Android. Existen varias opciones para hacerlo, como PyMob o la librería pgs4a (Pygame Subset for Android). Es un lenguaje muy accesible y fácil de aprender, considerado como muy elegante por muchos desarrolladores, por lo que puede ser una opción interesante a la hora de **elaborar aplicaciones sencillas**, porque para cosas complejas lo cierto es que no es muy recomendable porque hay que considerar que se pierden funcionalidades en comparación con el desarrollo en Android Studio.

B4A



Si lo que estamos buscando es **simplicidad y rapidez** y, sobre todo, no tener que lidiar con Java, B4A es lo que necesitamos, pues es en esos factores en los que se enfoca esta herramienta. B4A significa «Basic for Android» y es un IDE que permite desarrollar aplicaciones utilizando el lenguaje BASIC (Beginners All Purpose Symbolic Instruction Code), que es muy fácil de aprender pues prácticamente se lee como si fuese inglés.

Lo interesante de B4A es que **no sacrifica ninguna funcionalidad** a la hora de cumplir con lo que promete. Se puede acceder a las mismas API y librerías que en Android Studio, de manera que con un poco de creatividad se pueden hacer muchas cosas. Otra ventaja es que el código que usemos se puede trasladar con facilidad a B4i

<https://www.b4x.com/b4i.html> de manera que podamos generar sin muchas complicaciones también una aplicación para iOS. Por otro lado, en Android Studio es más fácil construir aplicaciones con Material Design que en B4A y, como es lógico, se pueden aprovechar mejor las nuevas características en el momento en que son introducidas.

## Unity





Terminamos esta lista con un entorno **especializado en un campo concreto: los juegos**. Unity es tanto un motor como un entorno de desarrollo que nos permite crear juegos multiplataforma de todo tipo: desde cosas muy simples hasta muy complejas con física realista, iluminación dinámica y gráficos en 3D. La capacidad de añadir código en C# o Java nos brinda además toda la flexibilidad que necesitemos para poder ir más allá de lo que el IDE nos ofrece por sí mismo. Incluso es posible construir aplicaciones que no sean juegos, aunque esto tiene un poco más de complejidad- Muchos de los juegos más exitosos de la Play Store han sido elaborados con Unity. A menos que quieras hacer algo muy específico y diferenciado, Unity ahorra la necesidad de tener que construir un motor desde cero, con todo lo que esto significa. Dado el éxito que las plataformas móviles han tenido en lo que se refiere a los juegos, Unity se ha convertido en uno de los **principales referentes** en cuanto al desarrollo y en una opción preferencial dentro de este campo.