

# Por qué trabajar con Maven y Gradle

---

Ya desde hace unos años todo el mundo utiliza un **gestor** para construir un proyecto y obtener sus **dependencias**. Ahora ya no podemos vivir sin estas tecnologías, pero antes se tenían que importar todas las dependencias a mano y meter a una persona en un proyecto se podía convertir en un infierno hasta que tenía todo el entorno configurado para empezar a trabajar. Hoy por hoy, entrar en un proyecto debería ser tan fácil como descargarlo y compilarlo para tener todo funcionando sin tener que perder mucho tiempo en su configuración.

Hay dos gestores que se han convertido en los más utilizados: Maven y Gradle.

## Maven

---

Maven es una herramienta de **software libre** para gestionar proyectos Java lanzada en 2002. Se basa en ficheros xml y viene con objetivos definidos para realizar ciertas tareas como la compilación del código y la generación del proyecto empaquetado, pero Maven también se encarga de obtener todas las dependencias del proyecto como de subirlo al repositorio final para que otros proyectos que dependan de él puedan usarlo.



Además, Maven se puede usar con multitud de añadidos aportados por la comunidad de desarrolladores, por ejemplo, hay **plugins** que se pueden usar para funciones como desplegar el proyecto en un servidor, pasar el proyecto a un analizador de calidad de código o generar la cobertura de test que tiene la aplicación.

## Gradle

---

Gradle también es una herramienta de **software libre** lanzada en 2007 pero sirve para gestionar proyectos Java, Groovy o Scala, aunque ya soporta otros lenguajes. Esta herramienta se configura con ficheros DSL basados en Groovy, en vez de los xml que usaba Maven.



Gradle ha surgido para solucionar el problema de la gestión de la construcción, dependencias y despliegue de los proyectos que tengan otro lenguaje diferente a Java, aunque este también soporta Java. Día a día va creciendo el uso de esta tecnología y al igual que Maven, tiene una gran cantidad de plugins que se pueden añadir para multitud de tareas relacionadas con la construcción, la gestión de dependencias y el despliegue.

## Conclusiones

---

El desarrollo sin Maven o Gradle debería ser una constante, usar una tecnología como estas en todos los proyectos. Además, estos proyectos estén bien configurados de forma que si hay que rotar a una persona en un proyecto, cuando entre el nuevo desarrollador pueda tener todo el entorno configurado en su equipo en el menor tiempo posible y por otro lado, estas tecnologías, también ayudan a que si un desarrollador tiene que cambiar de equipo, pueda tener el proyecto configurado en muy poco tiempo.