

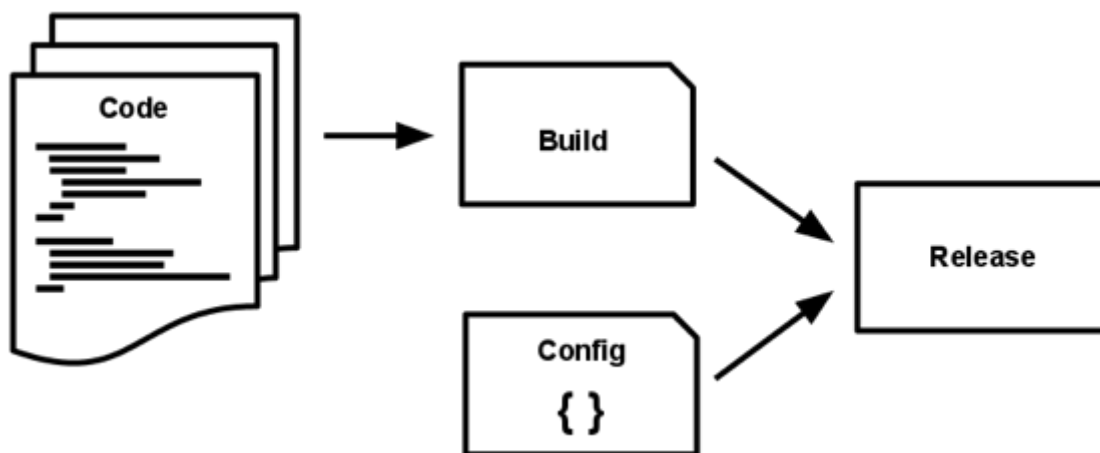
LA APLICACIÓN DE LOS DOCE FACTORES

V. Construir, liberar, ejecutar

Estrictamente separados construir y ejecutar etapas

Una [base de código](#) se transforma en una implementación (sin desarrollo) a través de tres etapas:

- La *etapa de compilación* es una transformación que convierte un repositorio de código en un paquete ejecutable conocido como *compilación*. Usando una versión del código en una confirmación especificada por el proceso de implementación, la etapa de compilación busca [dependencias de los](#) proveedores y compila binarios y activos.
- La *etapa de lanzamiento* toma la compilación producida por la etapa de compilación y la combina con la [configuración](#) actual de la implementación. La *versión* resultante contiene tanto la compilación como la configuración y está lista para su ejecución inmediata en el entorno de ejecución.
- La *etapa de ejecución* (también conocida como "tiempo de ejecución") ejecuta la aplicación en el entorno de ejecución, al iniciar un conjunto de [procesos](#) de la aplicación en una versión seleccionada.



La aplicación de doce factores utiliza una separación estricta entre las etapas de compilación, lanzamiento y ejecución. Por ejemplo, es imposible realizar cambios en el código en tiempo de ejecución, ya que no hay forma de propagar esos cambios a la etapa de compilación.

Las herramientas de implementación suelen ofrecer herramientas de administración de versiones, especialmente la capacidad de retroceder a una versión anterior. Por ejemplo, la herramienta de implementación de [Capistrano](#) almacena versiones en un subdirectorio llamado `releases`, donde la versión actual es un enlace simbólico al directorio de versión actual. Su `rollback` comando hace que sea fácil retroceder rápidamente a una versión anterior.

Cada versión siempre debe tener una ID de versión única, como una marca de tiempo de la versión (como 2011-04-06-20:32:17) o un número incremental (como v100). Las versiones son un libro mayor de solo anexo y una versión no se puede mutar una vez que se crea. Cualquier cambio debe crear una nueva versión.

Builds are initiated by the app's developers whenever new code is deployed. Runtime execution, by contrast, can happen automatically in cases such as a server reboot, or a crashed process being restarted by the process manager. Therefore, the run stage should be kept to as few moving parts as possible, since problems that prevent an app from running can cause it to break in the middle of the night when no developers are on hand. The build stage can be more complex, since errors are always in the foreground for a developer who is driving the deploy.

[ภาษาไทย \(th\)](#) | [한국어 \(ko\)](#) | [Slovensky \(sk\)](#) | [Русский \(ru\)](#) | [Français \(fr\)](#) | [日本語 \(ja\)](#) | [简体 中文 \(zh_cn\)](#) | [Ελληνικά \(el\)](#) | [Portugués brasileiro \(pt_br\)](#) | [Italiano \(it\)](#) | [Inglés \(en\)](#) | [Turco \(tr\)](#) | [Deutsch \(de\)](#) | [Polski \(pl\)](#) | [Українська \(reino unido\)](#) | [Español \(es\)](#).

«[Anterior](#)
[Próximo](#)»

[ภาษาไทย \(th\)](#) | [한국어 \(ko\)](#) | [Slovensky \(sk\)](#) | [Русский \(ru\)](#) | [Français \(fr\)](#) | [日本語 \(ja\)](#) | [简体 中文 \(zh_cn\)](#) | [Ελληνικά \(el\)](#) | [Portugués brasileiro \(pt_br\)](#) | [Italiano \(it\)](#) | [Inglés \(en\)](#) | [Turco \(tr\)](#) | [Deutsch \(de\)](#) | [Polski \(pl\)](#) | [Українська \(reino unido\)](#) | [Español \(es\)](#).

Escrito por Adam Wiggins

Última actualización 2017

[Código fuente](#)

[Descargar ePub Book](#)

[Política de privacidad](#)