

[TeamCity](#)

[Empezar a crear gratis](#)

Compilación y testeo Java con Gradle

Esta guía le enseña a compilar proyectos Java y Gradle con TeamCity, y se recomienda a desarrolladores totalmente nuevos en TeamCity.

Requisitos previos

Le recomendamos que tenga conocimientos básicos de Java y del marco de trabajo Gradle. Para obtener más información, consulte la [guía de introducción a Gradle](#) en la documentación de Gradle.

Paso 1: Crear un proyecto TeamCity

- 1.
2. Haga clic en + **Create Project** y seleccione la pestaña *From a repository URL*. En el campo *Repository URL*, introduzca su repositorio, por ejemplo: <https://github.com/marcobehlerjetbrains/teamcity-gradle.git>. TeamCity admite todos los sistemas de control de versiones populares sin configuración adicional: Git, Subversion, Mercurial, Perforce y TFS (TeamCity Cloud + On-Premises). La compatibilidad con CVS, StarTeam y Visual SourceSafe solo está disponible en TeamCity On-Premises.
3. Si su repositorio requiere autenticación, introduzca su nombre de usuario y contraseña o token de acceso.
4. Haga clic en **Proceed**.

Si TeamCity se conecta correctamente a su repositorio, verá el siguiente diálogo.

En el cuadro de diálogo *Create Project From URL* tiene la opción de cambiar el nombre del proyecto y el nombre de la configuración de compilación inicial.

Nota: En las versiones más recientes de TeamCity también verá los campos *Default branch* y *Branch specification*, que le permiten especificar qué ramas debe compilar TeamCity. Ignórellos por ahora.

- TeamCity le sugerirá un nombre de proyecto de forma predeterminada, pero puede elegir un nombre más adecuado para su proyecto si lo desea.
- TeamCity también sugerirá un nombre de configuración de compilación predeterminado. Puede dejar el valor predeterminado aquí y cambiarlo más tarde si lo necesita. (Cada proyecto de TeamCity consta de al menos una [configuración de compilación](#), que contiene todos los pasos necesarios para compilar su proyecto. Las configuraciones de compilación de TeamCity suelen llamarse *jobs* (trabajos) en otros sistemas de CI.)
- Haga clic en **Proceed**.

Al hacer clic en **Proceed**, TeamCity escanea automáticamente su repositorio de control de versiones en busca de tecnologías compatibles, en este caso Java y Gradle.

Si TeamCity detecta un archivo *build.gradle* en su repositorio, sugerirá automáticamente un [paso de compilación](#) para su proyecto que implique la compilación de su proyecto Gradle y la ejecución de sus pruebas mediante un `gradle clean build`.

Los pasos de compilación no deben confundirse con una configuración de compilación. Una configuración de compilación puede contener muchos pasos de compilación.

1. Seleccione la casilla situada junto al paso de compilación de *Gradle*.
2. Haga clic en **Use selected**.

Ya ha configurado correctamente su repositorio Gradle con TeamCity:

Paso 2: Ejecutar su primera compilación

Ya puede ejecutar sus primeras compilaciones.

1. Haga clic en el botón **Run** en la esquina superior derecha de la ventana, como se muestra aquí:

Nota: Si está utilizando TeamCity Cloud, podría tardar hasta un par de minutos hasta que un agente de compilación esté disponible. Durante este tiempo, su compilación esperará en la cola hasta que un agente disponible la recoja.

Si está usando TeamCity On-Premises con agentes de compilación locales, su compilación comenzará inmediatamente.

Cuando se inicie su compilación, se le redirigirá a la página de resumen de la compilación con la pestaña *Build Log* abierta, que muestra en tiempo real los datos relacionados con su compilación. Una vez que la compilación

haya terminado de ejecutarse, puede echar un vistazo a los [resultados de las pruebas](#) o navegar por el [registro de compilación](#) completo.

Paso 3 - Configurar su proyecto Gradle TeamCity

Ahora que su repositorio Gradle está conectado a TeamCity, puede continuar desarrollando y lanzando su código a su repositorio.

[De forma predeterminada](#), TeamCity sondeará la rama *principal* de su repositorio VCS **cada 60 segundos** en busca de cambios entrantes y desencadenará una compilación (combinada) para todas las confirmaciones detectadas.

Creación de ramas

Si desea activar una compilación por cada cambio en cualquier rama de su repositorio, no solo en la rama *principal*, añada una [especificación de rama comodín](#) a su configuración de VCS Root. Tenga en cuenta que las configuraciones VCS pertenecen al proyecto TeamCity, no a una única configuración de compilación. Por lo tanto, cualquier cambio que realice se aplicará a todas las configuraciones de compilación que utilicen la misma raíz VCS.

1. En la página de resumen del proyecto, haga clic en **Edit project**. Como alternativa, si tiene abierta la configuración de compilación *Build*, haga clic en **Edit Configuration**.
2. Navegue a *VCS Roots* (Version Control Systems) y edite su VCS Root.
3. Rellene el campo *Branch Specification* y haga clic en **Guardar**. Si no puede ver el campo *Branch Specification*, haga clic primero en **Show Advanced**.

Ejemplo de especificaciones de rama:

- `+:refs/heads/*`: TeamCity comprobará los cambios en todas las ramas de sus proyectos, pero no comprobará las solicitudes de incorporación de cambios en plataformas como GitHub, ya que coinciden con `refs/pull/*`.
- `+:*`: TeamCity comprobará cualquier cambio entrante en cualquier rama.
- Su propia especificación de rama personalizada.

TeamCity supervisará todas las ramas que se ajusten a su especificación de rama y se lancen a su repositorio en busca de cambios entrantes, y ejecutará compilaciones según corresponda.

Crear solicitudes de incorporación de cambios

Si desea que TeamCity compile automáticamente las solicitudes de incorporación de cambios a su repositorio, puede añadir la [función de compilación de solicitudes de incorporación de cambios](#) a su configuración de compilación.

1. Abra su configuración de compilación y haga clic en **Edit Configuration**.
2. Vaya a *Build Features* y haga clic en **Add Build Feature**. Si no puede ver el enlace **Build Features**, haga clic en **Show More**.
3. Seleccione **Pull Requests** en la lista desplegable y elija su repositorio, así como el proveedor de repositorios (GitHub, GitLab o similar).

4. Opcionalmente, aplique **Pull Request Filtering**, por autor o nombre de rama.

Nota: La función de compilación de *Pull Request* amplía de forma transparente la especificación de la rama (véase el paso anterior para más información). Por ejemplo, en el caso de GitHub, la función de solicitud de incorporación de cambios añadirá (de forma invisible) `+:refs/pull/*` a su especificación de rama.

Recomendamos que se asegure de que las ramas de solicitud de incorporación de cambios no se incluyan en su especificación de rama general cuando se utilice la función de solicitud de incorporación de cambios, porque de lo contrario las funciones relacionadas con la solicitud de incorporación no estarán disponibles en TeamCity.

TeamCity comprobará ahora la plataforma externa en busca de solicitud de incorporación de cambios y lanzará una compilación para aquellos que coincidan con sus reglas de configuración.

Nota: Debe utilizar esta función con precaución en los repositorios públicos, ya que cualquiera podría introducir código dañino en el repositorio (que usted no querría compilar).

Commit Status Publisher

Cuando se utiliza la función de solicitudes de incorporación de cambios en combinación con [Azure DevOps](#), [Bitbucket Server](#), [GitHub](#) o [GitLab](#), también tiene sentido utilizar la [función de compilación Commit Status Publisher](#). Esta función actualizará el estado de la solicitud de incorporación de cambios en la plataforma correspondiente con los resultados de la compilación.

Para configurar TeamCity para que informe de los resultados de la compilación a GitHub, tendrá que seguir estos pasos:

1. Abra su configuración de compilación y haga clic en **Edit Configuration**.
2. Vaya a *Build Features* y haga clic en **Add Build Feature**.
3. Seleccione **Commit Status Publisher** en la lista desplegable y elija su repositorio y un editor (GitHub, GitLab o similar).
4. Indique un token de acceso con derechos suficientes para publicar estados de confirmación.
5. Haga clic en **Save**.

Después de que TeamCity ejecute una compilación, ahora podrá ver fácilmente si los cambios causaron un fallo de compilación directamente desde la pestaña *Pull Request* en GitHub (marca de verificación verde).