

translated by Google

Se usó la API de Cloud Translation

([//cloud.google.com/translate/?hl=es-419](https://cloud.google.com/translate/?hl=es-419)) para traducir esta página.

Switch to English

Conceptos básicos de la integración continua

La integración continua (CI) es una práctica de desarrollo de software en la que los desarrolladores combinan con frecuencia sus cambios de código en un repositorio central, después de lo cual se ejecutan compilaciones y pruebas automatizadas.

Puedes configurar un sistema de CI básico para evitar cambios nuevos que afecten la compilación después de combinarla. Puedes programar un sistema de CI más avanzado para probar automáticamente la app y asegurarte de que se comporte como se espera en diferentes entornos, como niveles de API, tamaños de pantalla y plataformas.

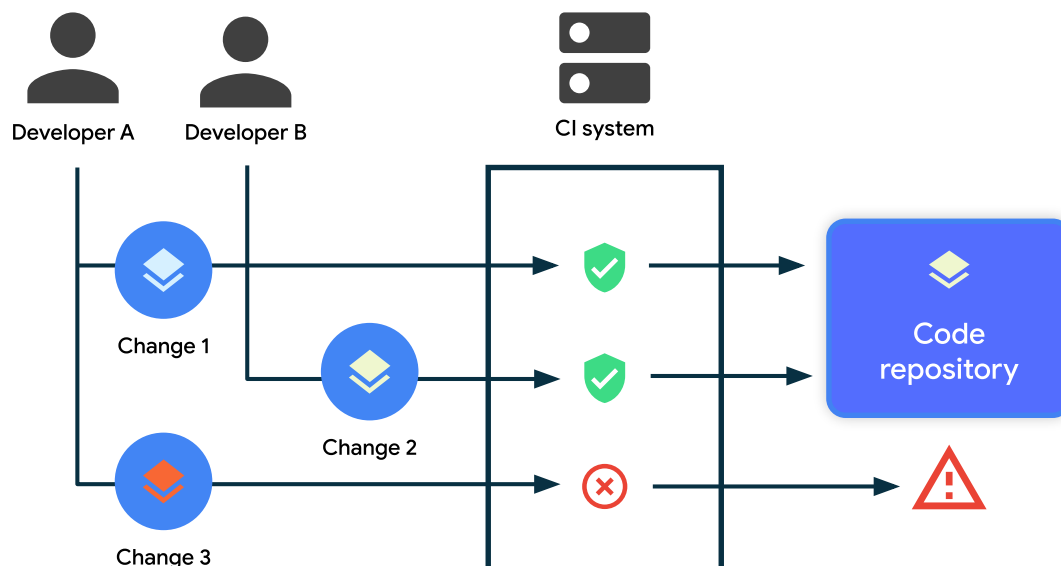


Figura 1. Un sistema de CI mantiene un repositorio de código en buen estado mediante la ejecución de verificaciones antes de la combinación.

En este documento, se muestran las estrategias comunes que usan los desarrolladores para configurar sistemas de CI eficaces para proyectos de Android. Estos lineamientos son genéricos y se aplican a la mayoría de las soluciones.

Ejemplo típico

Un sistema de CI típico sigue un *flujo de trabajo* o una *canalización*, que podría verse de la siguiente manera:

1. El sistema de CI detecta un cambio en el código, por lo general cuando un desarrollador crea una solicitud de extracción, también llamada “lista de cambios” o “solicitud de combinación”.
2. Aprovisiona y, luego, inicializa un servidor para ejecutar el flujo de trabajo.
3. Recupera el código y las herramientas, como el SDK de Android o las imágenes del emulador, si es necesario.
4. Compila el proyecto mediante la ejecución de un comando determinado, por ejemplo `./gradlew build`.
5. Ejecuta un comando determinado, por ejemplo, `./gradlew test`, para ejecutar las pruebas locales (<https://developer.android.com/training/testing/local-tests?hl=es-419>).
6. Inicia los emuladores y ejecuta las pruebas de instrumentación (<https://developer.android.com/training/testing/instrumented-tests?hl=es-419>).
7. Sube artefactos, como resultados de pruebas y APKs.



Figura 2: Un flujo de trabajo básico de CI

Beneficios de la CI

Las ventajas de la CI incluyen las siguientes:

- **Calidad mejorada del software:** La CI puede ayudar a mejorar la calidad del software a través de la identificación y solución de problemas desde el principio. Esto puede ayudar a reducir la cantidad de errores en las versiones de software y mejorar la experiencia general del usuario.
- **Menos riesgo de compilaciones dañadas:** Cuando automatizas el proceso de compilación con CI, puedes evitar mejor las compilaciones dañadas si resuelves los problemas antes en el proceso.
- **Mayor confianza en las versiones:** La CI puede ayudar a garantizar que cada versión sea estable y lista para la producción. Mediante la ejecución de pruebas automatizadas, la CI puede identificar cualquier problema potencial antes de su lanzamiento al público.
- **Comunicación y colaboración mejoradas:** Debido a que proporciona un lugar central para que los desarrolladores compartan los resultados de las pruebas y el código, la CI puede facilitar que los desarrolladores y otros miembros del equipo trabajen juntos y hagan un seguimiento del progreso.
- **Mayor productividad:** La CI puede ayudar a aumentar la productividad de los desarrolladores mediante la automatización de tareas que, de otro modo, llevarían mucho tiempo y serían propensas a errores.

Lecturas adicionales

Si deseas obtener más información sobre cómo usar la integración constante para mejorar el desarrollo de tu app, consulta las siguientes páginas:

- **Automatización de CI**
(<https://developer.android.com/training/testing/continuous-integration/automation?hl=es-419>)
- **Funciones de CI**
(<https://developer.android.com/training/testing/continuous-integration/features?hl=es-419>)

El contenido y las muestras de código que aparecen en esta página están sujetas a las licencias que se describen en la [Licencia de Contenido](https://developer.android.com/license?hl=es-419) (<https://developer.android.com/license?hl=es-419>). Java y OpenJDK son marcas registradas de Oracle o sus afiliados.

Última actualización: 2023-12-01 (UTC)