

## **Branching strategies**

Este concepto se refiere a las estrategias que se utilizan cuando se desarrolla software y se está utilizando un sistema de control de versiones como Git, que permite a un equipo trabajar en el desarrollo de forma concurrente. Su objetivo es evitar o resolver de forma eficiente los problemas que se puedan presentar cuando el equipo trabaje en el mismo código fuente de un proyecto.

## **Mono-repo y un poly-repo**

En el caso de un mono-repo se refiere a utilizar un solo repositorio para el sistema de control de versiones del código. Entonces, por ejemplo un repositorio de este tipo puede tener una carpeta para un proyecto de una aplicación web y otra para una aplicación móvil.

Por otra parte, un poly-repo se refiere a usar múltiples repositorios, entonces, a diferencia del anterior aquí se utilizaría un repositorio para un proyecto de una aplicación web y otro repositorio para el proyecto de una aplicación móvil.

## **Git submodules**

Un git submodule es como un repositorio dentro de otro repositorio de git. Se podría decir que los submodules son como hijos de un repositorio padre. En este caso el repositorio padre apunta hacia un commit específico del submodule, esto le permite al padre tener acceso a todos los archivos de ese commit. Esto permite clonar otro repositorio en el proyecto principal y a su vez mantener los commits de cada uno por separado. Un punto que hay que considerar es que si el submodule cambia, ese cambio no se incorpora automáticamente en el padre, sino que hay que cambiar el puntero al nuevo commit de forma manual.

## **Git flow**

Este es un flujo de trabajo aplicado a un repositorio Git. Es una manera de manejar los branches en un repositorio (Branching strategy). Uno de los beneficios de este es que permite el desarrollo paralelo. Aquí hay cinco tipos diferentes de branches: Main, Develop, Feature, Release y Hotfix, las cuales van a permitir trabajar de forma ordenada en un proyecto dentro de un repositorio.

## **Upstream y el remote para un repositorio local de git**

En el caso que se quiera hacer contribuciones a repositorios públicos externos o realizar una copia de algún repositorio remoto por lo general se hace un Fork de ese repositorio remoto. En este escenario, Upstream se podría decir que es el branch seguido en el repositorio remoto por la rama en repositorio local. A partir del Upstream se puede traer los cambios desde el repositorio remoto al local, ya sea para aprovecharlos o resolver conflictos a nivel local para posteriormente realizar un pull request al repositorio remoto. Con el comando remote se puede crear, ver y eliminar conexiones con otros repositorios.

## **Pull request**

Cuando se quiere hacer contribuciones a un repositorio remoto, primero se hace un Fork para copiar el repositorio original, y después de hacerle cambios se debe hacer un pull request para solicitar al propietario del repositorio original que incorpore los commits que están en el fork.

## **Ventaja de las soluciones de manejo de código como: Github, Gitlab, Bitbucket**

Entre las principales ventajas del uso de este tipo de solución están: Permite trabajar en equipo de forma paralela, permite trabajar sin conexión a la red por medio del repositorio local, brindan autonomía a los desarrolladores, permite la colaboración entre diferentes proyectos y desarrolladores, son multiplataforma, son sistemas escalables.