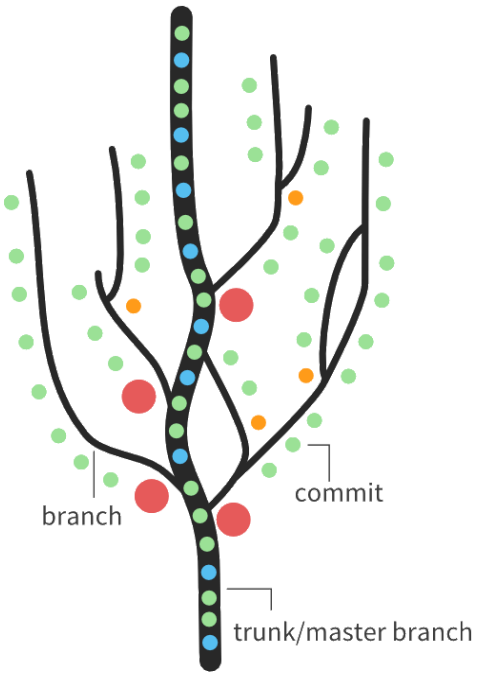
Material descargable

egg

GIT: Control de Versiones

¿Qué es una rama de Git?

Imagina tu proyecto como un árbol con muchas ramificaciones. La rama principal, conocida como "master" o "main", contiene la versión estable de tu proyecto. Las ramas en Git te permiten crear "caminos paralelos" para experimentar, desarrollar nuevas características o solucionar errores, sin alterar la rama principal.



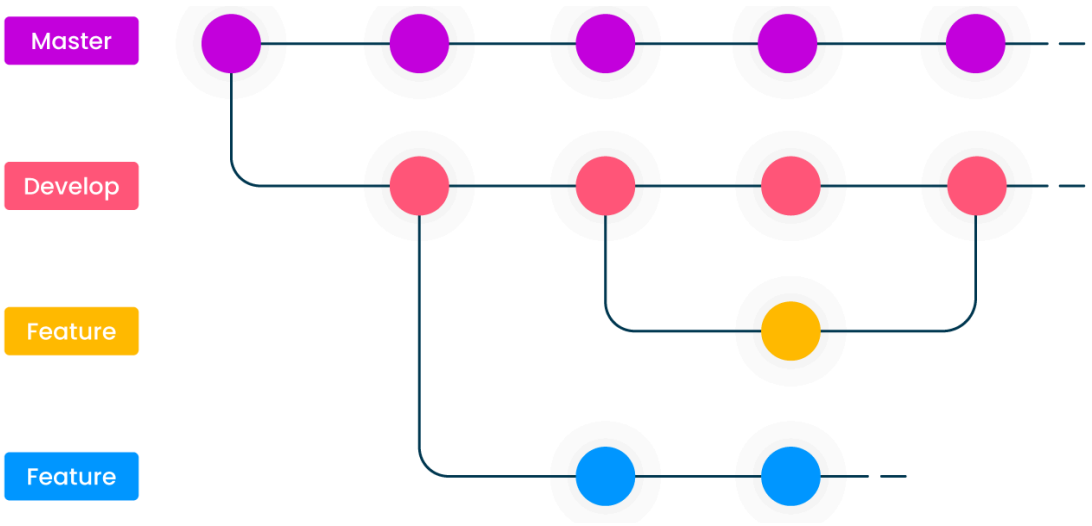
Al crear una rama, efectivamente haces una copia del proyecto en un punto específico del tiempo. Puedes trabajar en esta rama de manera aislada, probando nuevas ideas y haciendo cambios sin afectar la rama principal. Una vez satisfecho, puedes fusionar esta rama con la principal para integrar tus mejoras o correcciones. ¡Es como viajar en el tiempo en el mundo del desarrollo!

Las ramas facilitan el trabajo en equipo, permitiendo que cada miembro trabaje en distintas características o correcciones de forma independiente.

¿Qué es el git workflow?

El flujo de trabajo en Git, o Git Workflow, es un conjunto de prácticas recomendadas para trabajar de manera eficiente y productiva. Este flujo define cómo añadir, probar y fusionar nuevas funcionalidades de forma que se eviten conflictos y problemas. Un buen flujo de trabajo ofrece:

- Facilidad para deshacer cambios y corregir errores.
- Escalabilidad, adaptándose tanto a equipos pequeños como grandes.
- Claridad y simplicidad, permitiendo que todo el equipo trabaje de manera coherente.



¿Cómo se planifica este flujo de trabajo?

Veamos un ejemplo. Ana y Carlos están trabajando en un proyecto de desarrollo de software. Tienen una rama principal llamada **main** que contiene la versión estable del proyecto.

Ambos acuerdan trabajar en diferentes características. Ana trabajará en una nueva funcionalidad de búsqueda, mientras que Carlos se enfocará en mejorar la interfaz de usuario.

Ana en su computadora crea una rama desde main llamada **busqueda** para su funcionalidad, mientras que Carlos desde la suya, crea una rama llamada **mejora-interfaz**

Los dos trabajan en sus respectivas ramas, realizando **commits** regularmente. Esto les permite desarrollar y probar sus cambios sin afectar la rama main.

Periódicamente, Ana y Carlos **actualizan** sus ramas con los últimos cambios en la **nube** de la rama **main** para evitar futuros conflictos.

Una vez completadas las tarea, ambos suben sus archivos nuevos y/o modificados al **repositorio remoto** para luego, revisarlos, corregirlos, aprobarlos e integrarlos en la rama **main**.

Ana y Carlos continúan trabajando en nuevas características y mejoras, repitiendo este ciclo. Mantienen una **comunicación** constante y sincronizan sus ramas con main regularmente.

¿Cuáles son los comandos para trabajar con ramas?

Los comandos que nos ayudarán con esta tarea son::

git branch: Este comando se utiliza para listar todas las ramas locales del repositorio local actual.

git branch nueva-rama: Crea una nueva rama local llamada “nueva-rama”.

git branch -D nueva-rama: Borra la rama local llamada “nueva-rama”.

git checkout nueva-rama: Cambia la rama actual a la rama “nueva-rama”.

git checkout -b nueva-rama: Crea una nueva rama llamada nueva-rama y cambia a ella inmediatamente.

git switch nueva-rama: Este es un comando más reciente e intuitivo para cambiar de una rama a otra. Funciona de manera similar a git checkout nueva-rama pero está específicamente diseñado para cambiar de rama.

git push origin nueva-rama: Sube la nueva-rama al repositorio remoto en “la nube”. Este comando es útil para compartir la rama con otros colaboradores o para guardar el progreso en el repositorio remoto.

Cada uno de estos comandos es una herramienta esencial en el flujo de trabajo de Git, permitiendo a los desarrolladores gestionar eficientemente las diferentes líneas de desarrollo en sus proyectos.