

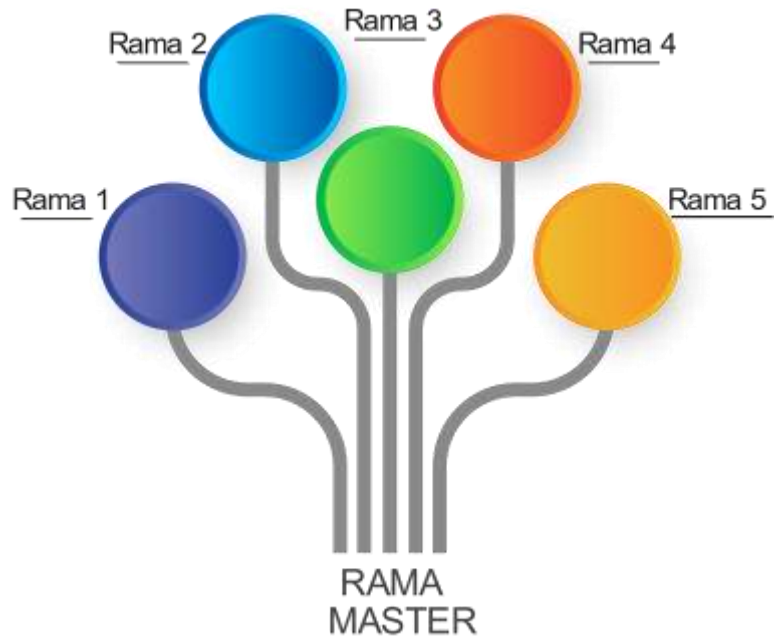


RAMAS EN GIT

En un proyecto colaborativo, es indispensable contar con un adecuado control de cambios, por lo que el uso de ramas permite aplicarlo en cada aspecto del sistema.

Git nos permite crear N cantidad de ramas según lo requiera nuestro proyecto, podemos manejarlas de acuerdo con nuestras necesidades, es decir podemos crear una rama exclusiva para los cambios de front end, una rama específica para controladores, etc así se pueden evitar “errores de dedo”

o que elementos de un proyecto colaborativo que no tengan injerencia en un ámbito del desarrollo intervengan en acciones que no les correspondan.



Una de las ventajas de las ramas es que se pueden fusionar, para esto existen dos estrategias diferentes **Fast Forward** que es cuando los cambios se hacen de manera automática y **Rekursiva**, que lo que hace es que compara los cambios que hay en una rama y otra y los une en una rama destino.

En esta última pueden surgir conflictos, en los casos en que la misma línea posee cambios diferentes en cada rama, para esto se requiere decidir cuál de las dos modificaciones es la que se va a establecer como correcta y hacer un commit en la rama destino. Esto resulta sumamente útil con equipos colaborativos, pero carecen de una eficiente comunicación.

De igual manera nos permite el uso de etiquetas, esto es para identificación o control de versiones, lo cual también nos permite etiquetar no solo el commit actual si no todos los que tengamos en nuestra rama, con lo que facilita la búsqueda y obtención de la información.

Otra de las características más interesantes de las ramas es el comando stash, que nos permite guardar los cambios, pero que no sean visibles en el repositorio, es decir que para hacer el commit no sean incluidos, y podremos recuperarlos en cualquier momento.

VENTAJAS DE LAS RAMAS

- Facilitan la separación de tareas
- Mejoran el entorno colaborativo
- Hacen más sencillo el control de cambios
- Errores sustanciales no afectan el ambiente de producción
- Es más fácil la adición de colaboradores en las diferentes actividades del proyecto
- Permiten trabajar más de una cosa a la vez
- Evitan errores al subir cambios
- Evitan sobrescribir sobre el código ya realizado por otro elemento del equipo
- Permiten una fácil identificación de versiones
- Hacen más Fácil subir solo lo necesario al repositorio
- Las fusiones se hacen de manera rápida y Transparente
- Facilitan el trabajo en equipos con problemas de comunicación entre integrantes