

Manejo de ramas en GitHub 24/43

RECURSOS

MARCADORES

Si no te funciona el comando `gitk` es posible no lo tengas instalado por defecto.

Para instalar `gitk` debemos ejecutar los siguientes comandos:

```
sudo apt-get update
sudo apt-get install gitk
```

Las ramas nos permiten hacer cambios a nuestros archivos sin modificar la versión principal (master). Puedes trabajar con ramas que nunca envías a GitHub, así como pueden haber ramas importantes en GitHub que nunca usas en el repositorio local. Lo crucial es que aprendas a manejarlas para trabajar profesionalmente.

Si, estando en otra rama, modificamos los archivos y hacemos *commit*, tanto el historial(`git log`) como los archivos serán afectados. La ventaja que tiene usar ramas es que las modificaciones solo afectarán a esa rama en particular. Si luego de “guardar” los archivos(usando `commit`) nos movemos a otra rama (`git checkout otraRama`) veremos como las modificaciones de la rama pasada **no aparecen** en la `otraRama` .

Comandos para manejo de ramas en GitHub

- Crear una rama:
`git branch branchName`
- Movernos a otra rama:
`git checkout branchName`
- Crear una rama en el repositorio local:
`git branch nombre-de-la-rama` o `git checkout -b nombre-de-la-rama` .
- Publicar una rama local al repositorio remoto:
`git push origin nombre-de-la-rama` .

Recuerda que podemos ver gráficamente nuestro entorno y flujo de trabajo local con Git utilizando el comando `gitk` . `Gitk` fue el primer visor gráfico que se desarrolló para ver de manera gráfica el historial de un repositorio de Git.