Andrés Artavia-2017075875
Investigación 1 - SOA

Resumen

En cuanto a la primera pregunta, se refiere a la estrategia usada por equipos de trabajo para definir la creación y distribución de ramas. Eso ayuda a mantener separadas las diferentes secciones de trabajo y evitar conflictos y facilitar el combinar el trabajo de múltiples personas y evitar que se den demasiados conflictos.

La segunda, menciona el mono repo, que es tener un repositorio donde van a parar todos los proyectos, aunque no estén relacionados. El poly-repo, es lo contrario, es tener un repositorio separado por proyecto o módulo.

La tercera es sobre git submodules, en donde se investigó que es una manera de mantener módulos de un proyecto completo, en un solo repositorio pero separados. De tal forma que se pueda trabajar independiente en ellos, pero se puedan usar dentro del mismo repositorio.

En cuanto al git flow, es una forma de manejar repositorios, en donde se crean ramas de desarrollo sobre la master y conforme se agregan funcionalidades, se agregan más ramas sobre develop hasta terminar el trabajo y luego fusionarlo de vuelta a la rama develop de donde salieron. Luego cuando una rama develop tiene todo lo necesario, o lo necesario para una versión del proyecto, se hace merge sobre la master.

La quinta, se tiene el remote, que es el lugar donde se está manejando el repositorio, puede ser en la nube o a nivel local. El upstream, es el trabajo o cambios que se quieren subir al remote.

Pull request es una solicitud para hacer merge de una rama a otra. La idea es que haya personas que revisen los cambios y aprueben el merge, para evitar que se hagan merges con cosas que no fueron probadas o que no se quieren en la rama a la que se le quiere hacer merge.

En general las ventajas de usar un manejador de código son varias, la primera y más simple es el respaldo en la nube de un proyecto. Luego está la capacidad de mantener el historial de cambios en un proyecto y poder observarlos o poder volver a versiones anteriores. También la capacidad de hacer ramas, con diferentes versiones para mantener líneas de trabajo separadas y luego de manera fácil, combinar el trabajo de las personas.

Referencias

1. https://launchdarkly.com/blog/git-branching-strategies-vs-trunk-based-development/#:~:text=What%20is%20a%20branching%20strategy,their%20changes%20with%20each%20other.
2. https://kinsta.com/blog/monorepo-vs-multi-repo/#:~:text=The%20monorepo%20approach%20entails%20storing,code%20hosted%20in%20independent%20repositories.
3. https://git-scm.com/book/en/v2/Git-Tools-Submodules
4. https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow#:~:text=Gitflow%20is%20a%20legacy%20Git,software%20development%20and%20DevOps%20practices.
5. https://www.neonscience.org/resources/learning-hub/tutorials/git-setup-remote
6. https://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes