Anderson Taylor Cordero - 2014045517 - Investigación 1 - GIT - Branching Strategies - Git Flow

El presente trabajo tiene como objetivo el de establecer las bases necesarias sobre la herramienta Git, además de proveer al estudiante con conocimientos para los casos de uso complejos de la herramienta. A continuación se habla un poco de algunos conceptos que fueron investigados. El primer concepto es “branching strategies” y se refiere a la estrategia que usa un equipo de desarrollo de software al escribir, fusionar y enviar código en el contexto de un sistema de control de versiones como Git. Es muy utilizado para trabajar en equipo ya que todos lo hacen desde la misma base, pero cada quien en lo que le corresponde. Es importante también saber diferenciar entre “mono-repo” y “poly-repo”, donde el primero consta de un repositorio de código único donde coexisten varios proyectos en una estructura de directorio jerárquico. Acá cada biblioteca o archivo lib y cada aplicación contienen varios otros subdirectorios para albergar el código fuente en sí. En el caso del diseño poly-repo este distribuye el código entre múltiples repositorios. El grado en que la separación entre piezas lógicas de código puede variar de un idioma a otro y de un estilo a otro y cada repositorio se centra en una funcionalidad específica. Cuando se necesita código externo es donde brillan un submódulo de git, que es un registro dentro de un repositorio de host de git que apunta a una confirmación específica en otro repositorio externo. Estos se deberían usar si se necesita mantener una gestión de versiones estricta sobre sus dependencias externas, como cuando un componente externo o un subproyecto cambia demasiado rápido o los próximos cambios rompen la API. Otro caso es cuando se tiene un componente que no se actualiza con mucha frecuencia y desea rastrearlo como una dependencia del proveedor, y un tercer caso sería cuando se delega una parte del proyecto a una persona externa y se quiere integrar su trabajo en un momento o lanzamiento específico. Para el concepto de “Git Flow” se tiene que este es un modelo alternativo de creación de ramas en Git en el que se utilizan ramas de función y varias ramas principales. Se diferencia con el desarrollo basado en troncos en que “Gitflow” tiene diversas ramas de más duración y mayores confirmaciones, además que puede utilizarse en proyectos que tienen un ciclo de publicación programado. Un término muy importante es el de las ramas upstream, que se pueden definir como la rama seguida en el repositorio remoto por nuestra rama local (también llamada rama de seguimiento remoto). En cambio un control remoto en Git es un repositorio común que todos los miembros del equipo usan para intercambiar sus cambios. A diferencia de un repositorio local, un repositorio remoto normalmente no proporciona un árbol de archivos del estado actual del proyecto. Una acción muy importante y usada es Pull request, donde las solicitudes de extracción le permiten informar a otros sobre los cambios que ha enviado a una rama en un repositorio en GitHub. Una vez que se abre una solicitud de extracción, se pueden revisar los posibles cambios con los colaboradores y agregar confirmaciones de seguimiento antes de que sus cambios se fusionen en la rama base. Finalmente se encontró muchas ventajas en las soluciones de manejo de código como: Github, Gitlab, Bitbucket, que son sistemas de control de versiones de software que permiten mantener un histórico de las distintas versiones de un desarrollo. Consiste en almacenar copias de los ficheros que contienen el código fuente del proyecto en cada uno de sus estados. Entre sus ventajas se encuentra que permiten mantener un historial de todo el desarrollo del proyecto, además de añadir trazabilidad al desarrollo de software, ya que se puede ver qué cambios se han hecho en el código en cada versión. Muestran mucha información estadística de cómo se está desarrollando el proyecto (principales autores, número de versiones, cambios, etc.) y facilita mucho el trabajo en equipo.