Guía para trabajar colaborativamente en Git con ramas

Cuando trabajéis con Git como control de versiones en clase, seréis o bien administradores o bien usuarios.

A continuación, van a relatarse los pasos a seguir para reproducir el ejercicio que se expuso en clase entre un administrador (profesor) y un usuario (alumno).

Como profesor, lo primero que haremos será crear el proyecto en GitLab en el que queremos que nuestro alumno colabore.

Lo clonamos en nuestro ordenador (git clone + enlace que aparece en el botón ‘Clone with HTTPS’) y añadimos el contenido que creamos conveniente para inicializar el proyecto. Después añadimos los cambios a nuestro repositorio local, comiteamos los cambios y los subimos a nuestro repositorio remoto (git add . -> git commit “mensaje para comitear” -> git push origin master).

Como alumno, lo que haremos para tomar parte en el proyecto de nuestro profesor es hacer clic en el botón ‘Fork’ en el proyecto correspondiente de GitLab. A continuación, debemos elegir dónde vamos a trabajar en ese proyecto dentro de nuestro repositorio remoto, para ello, escogemos uno de los subgrupos que nos ofrece GitLab.

Una vez hemos terminado de hacer el fork, clonaremos nuestro proyecto en nuestro ordenador (git clone + enlace que aparece en el botón ‘Clone with HTTPS’).

En ese momento, el profesor añadirá un archivo con los trabajos que se le van a asignar a los alumnos en un documento de texto en su repositorio local. Para ello, copiará el archivo en la carpeta que corresponde con el repositorio local y ejecutará los siguientes comandos (git add . -> git commit “mensaje para comitear” -> git push origin master).

Como alumno, para mantener el repositorio sincronizado con el del profesor lo que haremos será añadir el remote de su repositorio remoto para bajar sus cambios. Para ello se ejecutará el comando git remote add profesor + enlace que aparece en el proyecto del profesor en el botón ‘Clone with HTTPS’

profesor será el nombre del nuevo remote. Si queremos comprobar los remotes que tenemos añadidos a nuestro repositorio local bastará con ejecutar el comando git remote.

Para terminar la sincronización y bajar los cambios del profesor habrá que ejecutar el comando git pull profesor master. Con esta orden pasan los cambios de remoto a local del proyecto al que apunta el remote profesor a la rama master de nuestro repositorio.

Como último paso de la sincronización ejecutamos el comando git push origin master para subir los cambios bajados del proyecto del profesor a mi repositorio remoto.

Lo que vamos a hacer a continuación es hacer el trabajo que ha encargado el profesor. Para ello, añadimos los archivos que sean necesarios con el trabajo indicado. En este caso, antes de añadirlos al repositorio local, comitearlos y pushearlos, lo que vamos a hacer es crear una nueva rama llamada cambios para realizar los cambios que nos ha pedido el profesor.

Cuando estamos en un proyecto, los cambios nunca han de estar en la rama master del proyecto, sino en una rama aparte. La rama master la reservaremos para tener sincronizado el proyecto con el del administrador correspondiente teniendo un proyecto funcional.

Para crear una nueva rama ejecutamos las siguientes órdenes: git branch cambios (con esto creamos una nueva rama) y git checkout cambios (con esto nos cambiaremos para trabajar a la rama cambios en lugar de master).

Una vez estamos en la rama correcta añadiremos los cambios (o trabajo que hemos realizado) como hacemos siempre (git add . -> git commit “mensaje con los cambios” -> git push origin cambios). Nótese que en este caso NO es git push origin master, sino git push origin cambios porque estamos subiendo nuestros cambios a la rama cambios.

Ya hemos hecho nuestro trabajo, lo hemos comiteado y subido a nuestro repositorio remoto. Lo que vamos a hacer ahora es pedirle al profesor que incorpore nuestros cambios al proyecto en el que estamos colaborando.

Como profesor, sabemos que los alumnos ya han hecho los trabajos asignados, entonces creamos tantas ramas como alumnos colaboradores tengamos para que nos envíen sus trabajos. Podemos crear las ramas necesarias estando en el proyecto en GitLab, en el panel de la izquierda, en Repositorio -> Ramas -> Nueva Rama. Nombramos la rama y clicamos en ‘Create Branch’.

Para colaborar en un proyecto del que hemos hecho un fork lo que hacemos es, en GitLab, en el proyecto que se creó cuando forkeamos el del profesor, es pulsar el botón ‘Crear una solicitud de fusión’. Una vez pulsado dicho botón, lo único que tenemos que modificar es el título de la solicitud y las ramas desde la que vamos a enviar y la rama a la que van a llegar los cambios.

Para cambiar las ramas pulsamos en el botón ‘Change branches’. Como rama desde la que mandamos los cambios elegiremos nuestra rama cambios y como rama de destino elegiremos la que el profesor haya creado para nosotros como alumno. Clicamos en Compare branches and continue y enviamos la solicitud.

Como profesor, vemos que hay alumnos que nos han mandado solicitudes de fusión en el panel de la izquierda de nuestro proyecto como ‘Merge Requests’. Aceptamos las solicitudes de los alumnos, y pasamos a revisar y aprobar los cambios que nos han mandado los alumnos.

Para hacerlo, en nuestro repositorio local creamos la rama del alumno del que vamos a revisar el trabajo. Digamos que cuando creamos la rama, la llamamos alumno. En ese caso ejecutaremos en local git branch alumno y posteriormente git checkout alumno para cambiarnos a la rama alumno y poder descargar los cambios que nos ha enviado. Para bajar los cambios del alumno ejecutaremos git pull origin alumno. Comprobamos que los cambios o el trabajo se ajustan a lo que hemos pedido y en el caso de que sea correcto, lo que vamos hacer es fusionar la rama alumno con la rama master de nuestro proyecto, donde están los archivos definitivos del proyecto.

Para fusionar las ramas lo primero es cambiarnos a la rama de destino, en este caso master con el comando git checkout master. Finalmente ejecutamos git merge alumno y fusionamos las ramas.

Lo único que nos falta como profesor es actualizar nuestro repositorio remoto con los cambios de los alumnos. Ejecutaremos git push origin master.

Recordad que los alumnos deben tener siempre la rama master de su repositorio sincronizada con la del profesor. Para hacer esto lo que hacemos es como alumnos asegurarnos de que estamos en la rama master y si no lo estamos, cambiar a ella con git checkout master. Lo último que se debe hacer es git pull profesor master y git push origin master para actualizar la rama donde tenemos el trabajo funcional.

Con esto finalizaría la colaboración y estaríamos preparados para hacer más cambios en el caso de que el profesor así nos lo pida.