

EJERCICIO 1

1. ¿Qué comando nos permite pasar cambios al Stage o hacer staging?

`git add`

2. ¿Con qué opción del comando `git commit`, indicamos el mensaje que queremos escribir en el commit?

`git commit -m`

3. ¿Con qué comando subimos un commit al repositorio remoto?

`git push`

4. ¿Cómo podemos crear una rama? Indicar las 2 formas:

Lo podremos hacer desde comandos o desde algún IDE o página.

4.1. cómo se haría en local por comando y subida al repo remoto.

Por comandos sería de la siguiente manera:

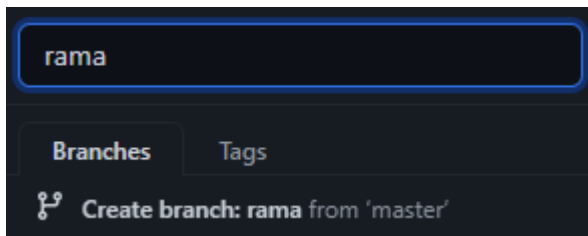
`git branch nombre_rama`

`git checkout -b nombre_rama`

Creamos la rama como se indica arriba y la subiremos con `git push`

4.2. y cómo se haría creando la rama en el repo remoto (github) y hacer una copia de la rama en el repo local.

Para crearla desde GitHub



Para traernos una rama remota

`git fetch origin nombre_rama:nombre_rama_local`

5. ¿Cómo puedo traerme los cambios que tenga en el repo remoto al repo local?

`git pull`

6. ¿En qué momento es recomendable hacer un commit del código?

Cuando el código sea funcional y la tarea que pretenda implementar esté completamente acabada

7. ¿Qué alternativa tenemos a hacer commit si quiero reservar los cambios no confirmados aún? Indica los comandos para gestionar esta alternativa.

Con git stash guardaríamos los cambios y con git stash pop los tendríamos de nuevo en el staging area

8. ¿Sería posible añadir ciertos ficheros a un commit y otros al stash? ¿Cómo lo harías? ¿En qué caso podría ser útil hacerlo?

Para subir ficheros a un commit lo haríamos con git add fichero1 fichero2 y posteriormente ejecutamos el comando git commit -m "texto" y por último, los ficheros no subidos en el commit los agregaremos al stash haciendo git stash.

Esto sería útil si estamos haciendo un desarrollo y nos surge un error en una implementación de otra rama.

9. ¿Cómo puedo saber las branches que tengo en local? Indica el comando.

git branch

10. ¿Es posible estar trabajando en varias ramas a la vez? ¿En qué caso podría ocurrir? ¿Cómo lo gestionarías el estar cambiando de una a otra rama?

Si estamos haciendo un desarrollo y nos surge un error en una implementación de otra rama.

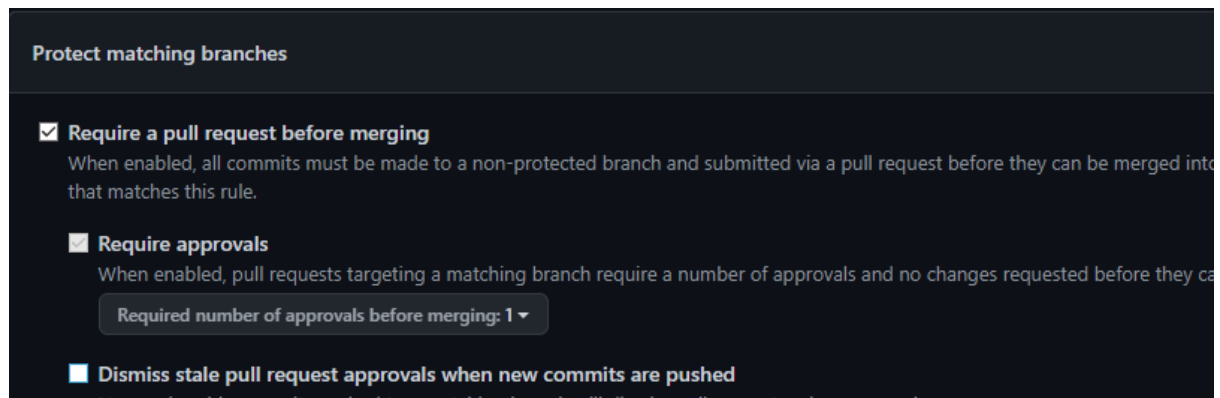
Si los cambios de la ram actual no están acabados haríamos un stash y con el comando git checkout nos cambiaríamos a la otra rama

EJERCICIO 2

En este ejercicio debéis demostrar que sabéis trabajar en equipo haciendo uso de Git. Las preguntas van a ir enfocadas a este tema.

1. ¿Se puede proteger la rama master en Github para que sea necesario la aprobación de un revisor? ¿Cómo se haría? Se pueden incluir capturas de pantalla. Explicar además si hubiera que tener en cuenta alguna configuración del repositorio.

Podremos establecer una política de seguridad desde el apartado de configuración del repositorio, en el apartado Branches



Para realizar estos cambios debemos tener en cuenta que el repositorio debe ser público

2. Estamos en el supuesto de que 2 desarrolladores están trabajando en el mismo repositorio y van a proteger la rama master para que sea necesario revisar y aprobar las ramas antes de mergear a master. Lo primero que hacen es crear 2 cards (tarjetas) en JIRA con la tarea que va a implementar cada desarrollador. Una de las cards creadas tiene el identificador APP-135 y será asumida por Mateo que es el desarrollador 1. La otra tarea tiene el identificador APP-180 y será implementada por Miguel. Se pide describir cómo debe ser el flujo que deben seguir los dos desarrolladores para trabajar en el mismo repositorio, cada uno en su rama, revisar el código mediante pull request y finalizar mergeando ambas ramas a master.

Ambos desarrolladores crearán una rama en su repositorio local identificada con el ID de la tarea asignada.

Una vez que la tarea se finalice, deberán de subir la rama y posteriormente solicitar una PullRequest de sus cambios.

El otro desarrollador comprobará si los cambios realizados contienen errores o son correctos. En el caso de que todo esté bien se mergeará la rama a máster, aunque habrá que tener en cuenta que si ambas tareas se realizan sobre el mismo fichero, el que más tarde suba los cambios deberá, previamente a mergear su rama a máster, mergear máster en su rama y solucionar los conflictos que pudieran surgir.

3. ¿Podrías explicar en qué caso se puede dar un conflicto trabajando de esta manera? ¿Cómo habría que resolverlo?

Como comentaba en la pregunta anterior si ambos desarrolladores trabajan sobre el mismo archivo pueden ocurrir conflictos a la hora de mergear la rama de desarrollo a la master, por lo que es recomendable actualizar en la rama los cambios hechos hasta ese momento en master, solucionando en dicha rama los posibles conflictos que pudiesen surgir.

Una vez dichos conflictos sean solucionados, mergearíamos la rama de desarrollo con master