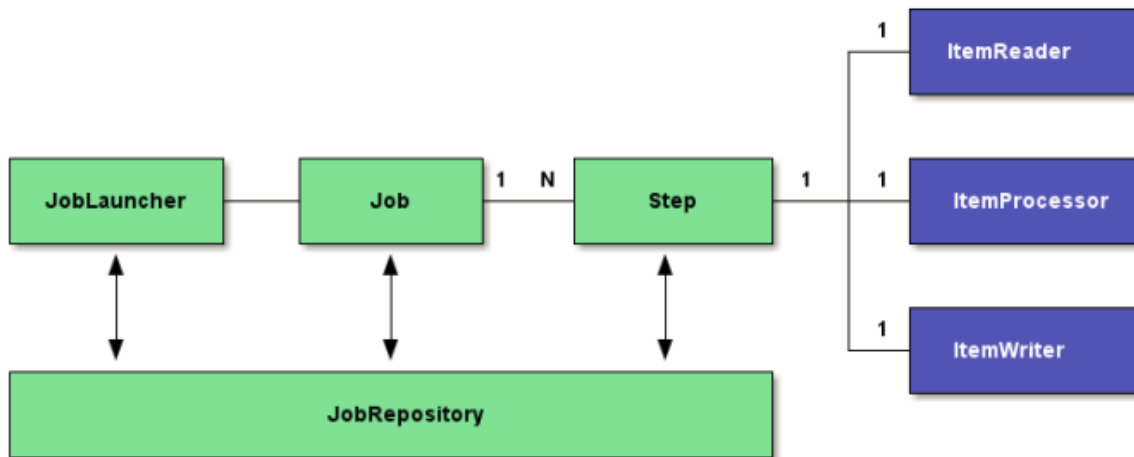


Becerril Beerril Daniel

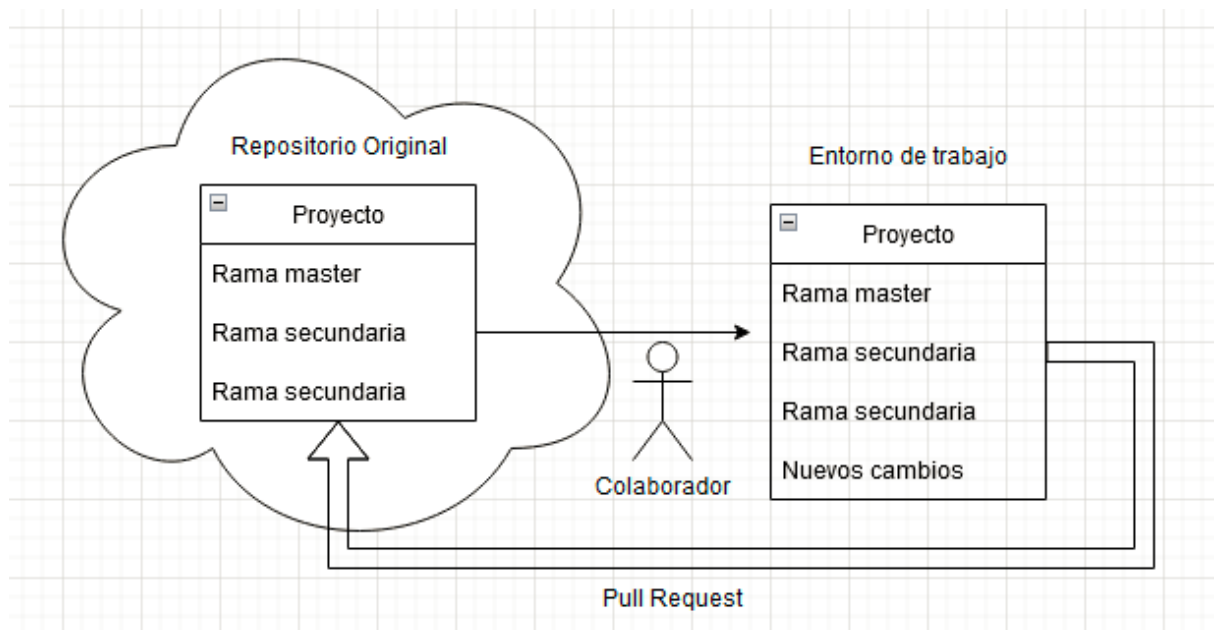
1. Diagramar el funcionamiento de Spring Batch.



El objetivo de Spring Batch es enfocarse en optimizar grandes tareas reduciendo el nivel de complejidad mediante la siguiente estructura, todo comenzará con el job launcher, quien será el encargado de iniciar todo el proceso una vez que se dispare un evento, este dará inicio a uno o más trabajos y cada uno puede tener uno o muchos pasos que se ejecutarán consecutivamente mediante 3 etapas, el inicio, el proceso y el cierre. El inicio es el itemReader, en el itemProcesor se realizan todas las acciones de procesamiento o lógica del negocio para lo que fue concebido ese proceso, aunque estas no son obligatorias, es decir que lo único obligatorio en los pasos de cada trabajo es el inicio y el cierre o itemWriter, ya que son los responsables de la entrada y la salida de los datos. El jobRepository es quien se encargará de la persistencia en las acciones de cada trabajo.

2. Explicar los comandos de git (pull request, fork, rebase, stash, clean, cherry-pick).

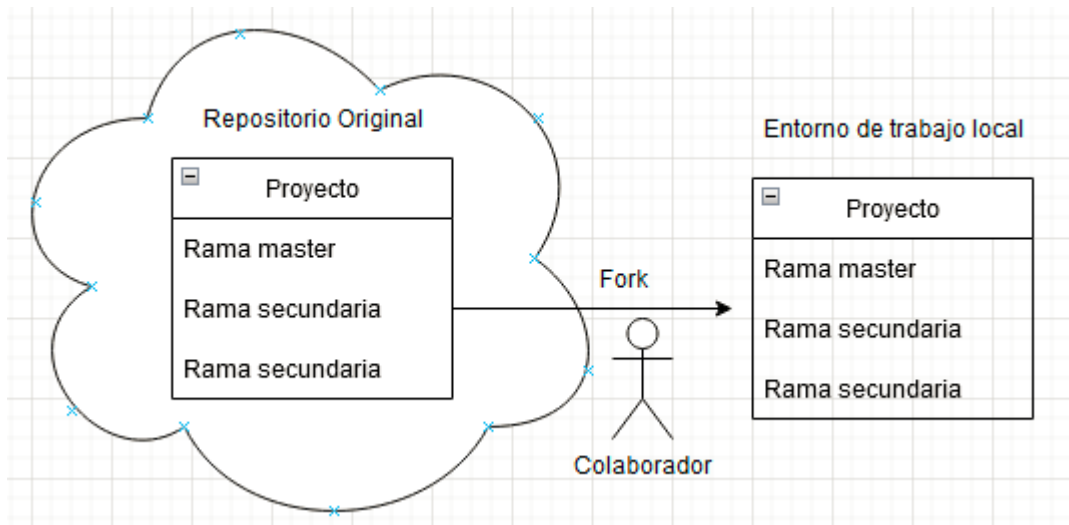
- Pull request.



Cuando se quiere hacer una solicitud a un repositorio se utiliza este comando para

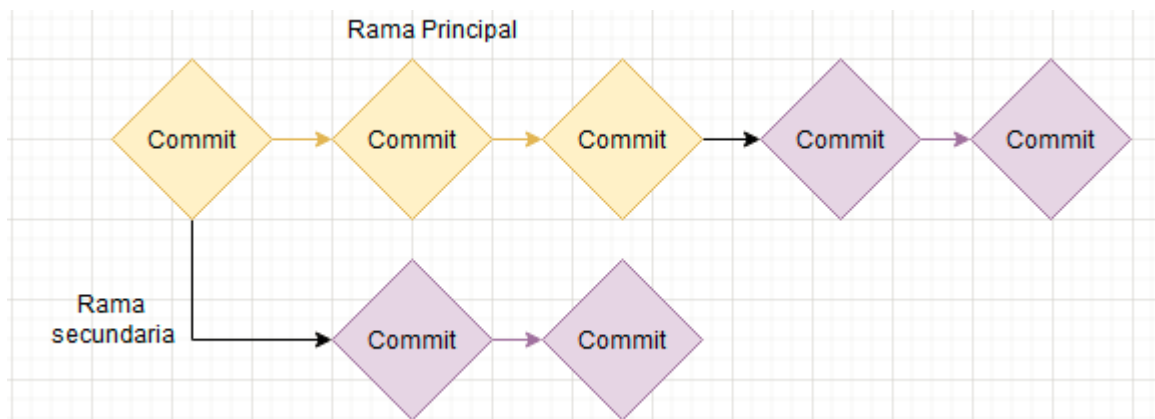
enviar cambios locales al servidor original e integrarlos en el proyecto.

- Fork.



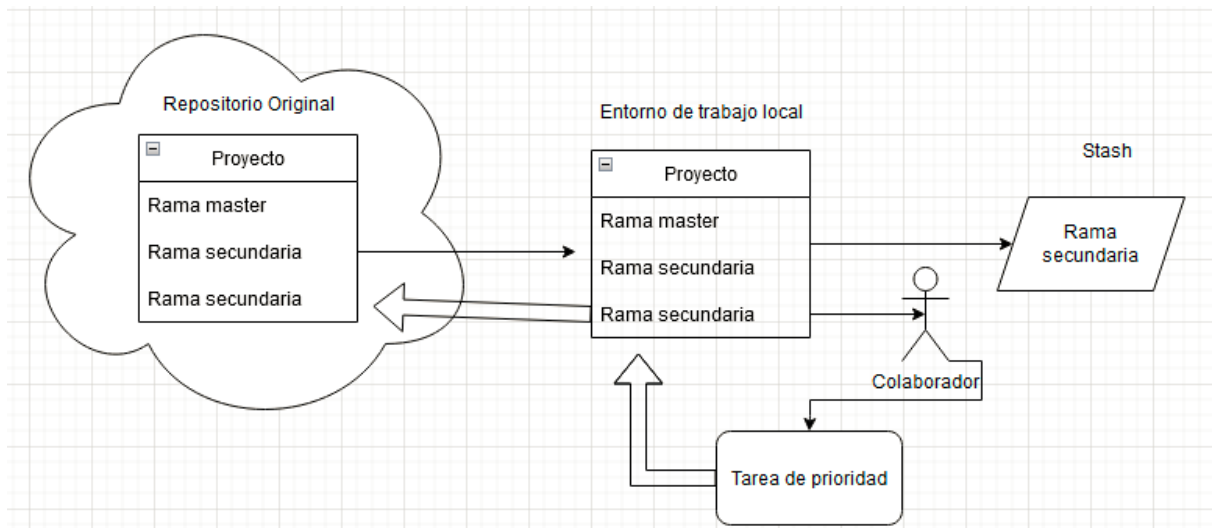
Un fork consiste en una especie de clonación del proyecto original hacia otro repositorio, sin embargo ambos repositorios puedes seguir conectados y conservando cierta relación, ya que todos los cambio que se hagan en el repositorio original después del fork se pueden llevar al repositorio clonado, es como una copia que se puede trabajar individualmente sin meterse con el repositorio original y aún asi conservar futuras actualizaciones.

- Rebase.



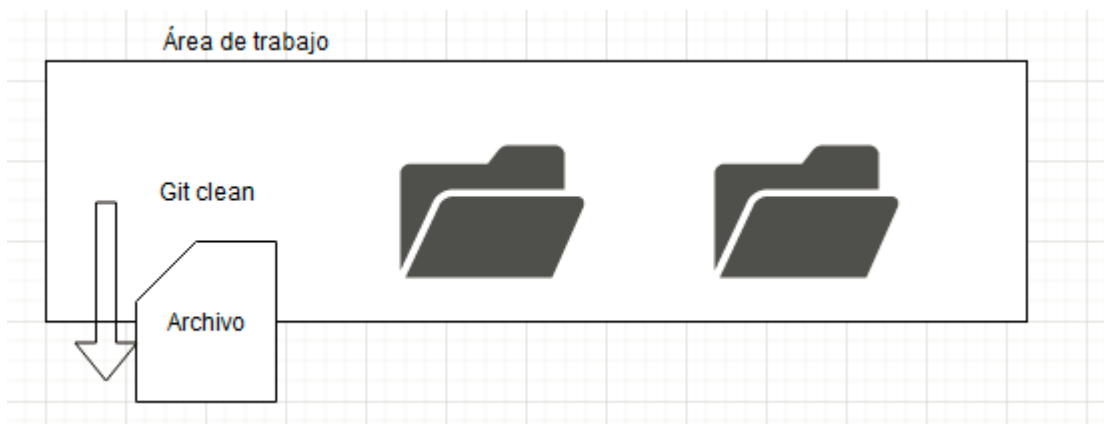
Este comando se utiliza para reorganizar la estructura de los commits trayendo y acomodando los de ramas exteriores, haciendo que se integren en la rama principal, modificando la base, no se debe usar este comando en producción porque puede alterar totalmente el trabajo de otras personas, pero es útil cuando no se ha enviado nada a producción y se ha trabajado tan desordenadamente que se perdió el orden.

- Stash.



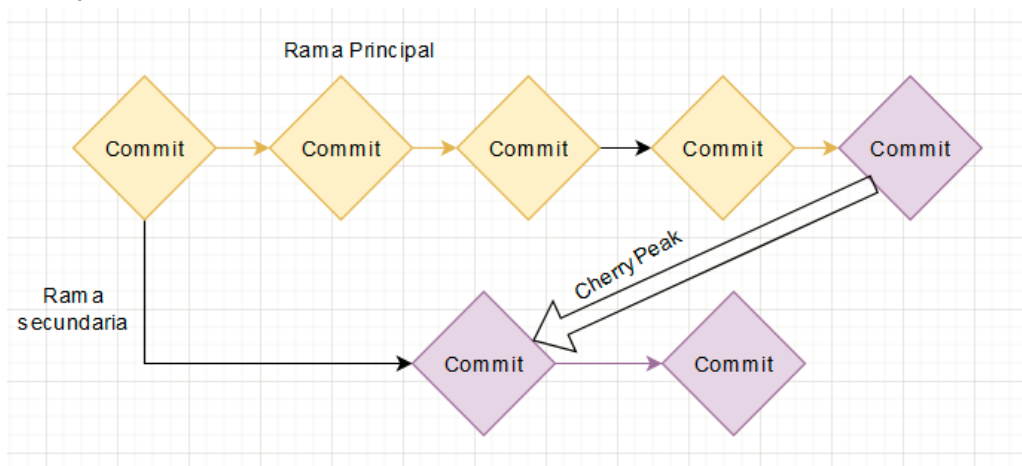
El stash es un espacio que protege el trabajo que no se quiere perder pero tampoco está listo para ser añadido con git add, por lo que es útil si necesitamos resolver cualquier otra tarea o trabajo sin perder nuestros avances en otro trabajo, la ventaja de esto es que no se integrará hasta que nosotros no lo recuperemos y lo integremos al repositorio.

- Clean.



Se utiliza clean para limpiar el área de trabajo de archivos o directorios que no queremos tomar en cuenta para llevar en el repositorio, de esta manera aunque estén presentes en la ruta ya no se tomarán en cuenta.

- Cherry-pick.



Este comando sirve para escoger un hito en la historia del proyecto con el que se quiere trabajar o integrar, la ventaja de esto es que permite traer una versión especificada por un commit en otra rama y traerla al directorio de trabajo. Esto es útil cuando se trabaja colaborativamente y alguna persona necesita trabajar sobre algo que está desarrollando otra persona.

3. Programa de aplicación front end.

[Repositorio](#)

4. Programa de Spring Batch.

[Repositorio](#)