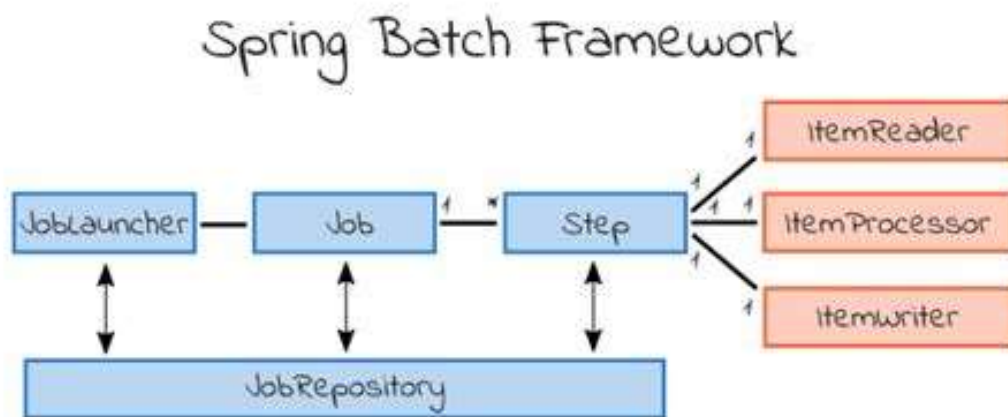


## 1- Crea el diagrama de los componentes de Spring Batch y explícalo.

Los procesos Batch (o procesos por lotes) son aquellos programas que se lanzan generalmente de manera programada y que no requieren ningún tipo de intervención humana. Suelen ser procesos relativamente pesados, que tratan una gran cantidad de información.

Spring Batch es un framework ligero enfocado específicamente en la creación de procesos Batch. Además de marcar unas directrices para el diseño de procesos, Spring Batch proporciona con una gran cantidad de componentes que intentan dar soporte a las diferentes necesidades que suelen surgir a la hora de crear estos programas



### Componentes:

**Job repository.** Es el componente encargado de la persistencia de metadatos relativos a los procesos tales como procesos en curso o estados de las ejecuciones. Almacena la información de cada Job por lotes y proporciona operaciones CRUD(Create, Read, Update y Delete) para el Job Launcher, Job y Step.

**Job Launcher.** Es el componente encargado de lanzar los procesos suministrando los parámetros de entrada deseados.

**Job.** El Job es la representación del proceso. Un proceso, a su vez, es un contenedor de pasos (steps) y un Job debe por lo menos tener un Step

**Step.** Un step (paso) es un elemento independiente dentro de un Job (un proceso) que representa una de las fases de las que está compuesto dicho proceso. Cada Step está formado por tres elementos que son ItemReader, ItemProcessor e ItemWriter.

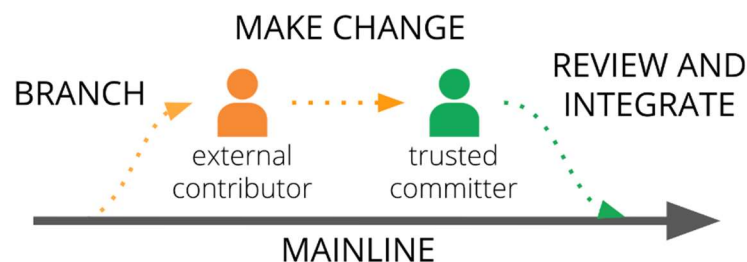
**ItemReader.** Elemento responsable de leer datos de una fuente de datos como BBDD, ficheros, cola de mensajes, entre otros.

**ItemProcessor.** Elemento responsable tratar la información obtenida por el reader. No es obligatorio su uso.

**ItemWriter.** Elemento responsable guardar la información leída por el reader o tratada por el processor. Si hay un reader debe haber un writer.

## 2- Explica los comandos de GIT: Pull request, Fork, Stash, Rebase, Cherry-pick y Clean.

**Pull request.** Un pull request es una petición que el propietario de un fork de un repositorio hace al propietario del repositorio original para que este último incorpore los commits que están en el fork.

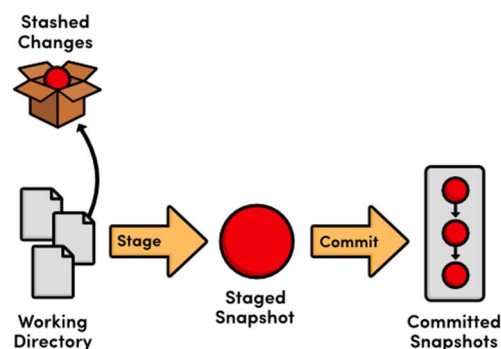


**Fork.** Un fork significa bifurcación, y en programación supone copiar un proyecto y partir de este, hacerle modificaciones. Cuando trabajamos con repositorios Git, supone hacer una copia exacta del proyecto, generando dos URL distintas.

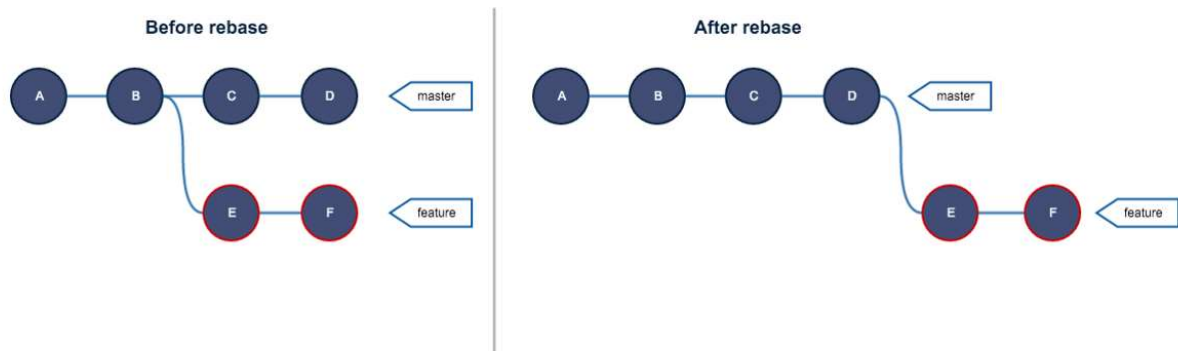


**Stash.** Es un área donde puedes almacenar temporalmente una captura de tus cambios sin enviarlos al repositorio. Está separada del directorio de trabajo (Working directory), del área de preparación (staging area), o del repositorio. Para guardar tus cambios en el stash, ejecuta el comando:

`git stash save "mensaje opcional para ti"`



**Rebase.** Realizar un rebase a una rama (branch) en Git es una forma de mover la totalidad de una rama a otro punto del árbol. El ejemplo más simple es mover una rama más arriba en el árbol

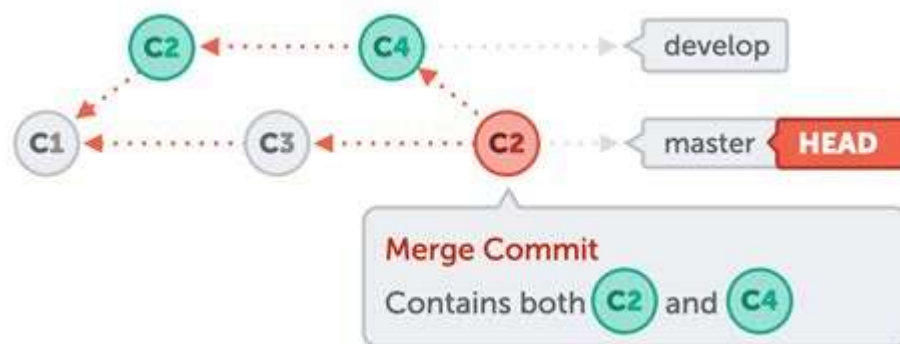


Para realizar rebase, asegúrate de tener todos los commits que quieras en el rebase en tu rama `master`. Revisar la rama en la que quieres hacer el rebase y escribe `git rebase master` (donde `master` es la rama en la que quieres hacer el rebase).

**Cherry-pick.** Se encarga de elegir uno o más commit o confirmaciones de cambios de una rama específica para luego aplicarla a otra rama.

Esto quiere decir que los commit de la plataforma de Git pueden elegirse por referencia, para que añadirlos al HEAD actual de trabajo, que no es más que el commit en el que se encuentra posicionado el repositorio del usuario en cada momento.

`git cherry pick "nombre del commit"` es recomendable solo en algunos casos ya que puede generar duplicados. se pueden utilizar también `git cherry pick --no-commit`.



**Clean.** Actúa en archivos sin seguimiento, este tipo de archivos son aquellos que se encuentran en el directorio de trabajo, pero que aún no se han añadido al índice de seguimiento de repositorio con el comando `add`. La ejecución del comando predeterminado puede producir un error. La configuración global de Git obliga a usar la opción `force` con el comando para que sea efectivo. Se trata de un importante mecanismo de seguridad ya que este comando no se puede deshacer.