GIT Avanzado

Fork: Función que permite crear una copia de un repositorio en una cuenta de usuario diferente. Sirve para contribuir a un proyecto de código abierto sin tener permisos para hacer cambios directamente en el repositorio original.
Pasos para crear un fork:

1. Ir a la página del repositorio al que se quiere hacer fork.
2. Hacer clic en el botón "Fork" en la esquina superior derecha.
3. Seleccionar la cuenta de usuario donde se quiere hacer el fork.
4. Una vez creado el fork, clonar el repositorio a una máquina local.

Muchas veces los fork se usan en combinación con el siguiente concepto.

Pull request: Característica en Git que permite solicitar a otras personas que revisen y acepten cambios realizados en un repositorio de código. Sirve para solicitar y aceptar contribuciones a un proyecto de manera colaborativa y controlada.
Pasos para hacer un pull request:

1. Hacer fork del repositorio original
2. Clonar el repositorio en el que se quieren hacer cambios.
3. Crear una rama para el cambio específico.
4. Realizar los cambios y hacer un commit con los cambios.
5. Hacer un push de la rama a la que se hicieron los cambios.
6. Ir a la página del repositorio en línea y hacer un pull request desde la rama modificada hacia la rama principal.
7. Uno o más revisores revisan los cambios y aceptan o rechazan el pull request.
8. Si es aceptado, el código es integrado en la rama principal.

Rebase: Técnica de integración de cambios que se utiliza para integrar una rama en otra. A diferencia de la integración por merge, en la que se crea un nuevo punto de merge que combina las dos ramas, el rebase permite integrar los cambios de una rama en la otra sin crear un nuevo punto de merge. En lugar de crear un nuevo punto de merge, Git "rebasa" los cambios de una rama sobre otra, esto es, integrar los cambios de una rama a otra.

Pasos para hacer un rebase:

1. Asegúrate de que estás en la rama que deseas integrar con otra.
2. Ejecuta el comando git rebase nombre_de_la_rama.
3. Git rebasará los cambios de la rama actual sobre la rama especificada.
4. Resuelve cualquier conflicto que pueda surgir durante el proceso.
5. Continúa el rebase con el comando git rebase --continue.

Stash: Permite guardar temporalmente los cambios no confirmados de una rama en un "stash" o "reserva". Sirve para cuando se desea cambiar de rama para trabajar en algo diferente, pero se quiere mantener los cambios no confirmados sin tener que hacer un commit temporal.

Pasos para hacer un stash:

1. Realiza cambios en tu rama local y verifica que tienes cambios no confirmados con git status.
2. Ejecuta el comando git stash para guardar los cambios no confirmados en un stash.
3. Cambia a otra rama si lo deseas.
4. Cuando estés listo para recuperar tus cambios, ejecuta git stash apply para aplicar los cambios guardados en el stash a la rama actual.

Clean: Permite limpiar la carpeta de trabajo de archivos que no están rastreados por el control de versiones. Es decir, permite eliminar archivos y directorios que no están siendo gestionados por Git. El comando git clean es una herramienta muy útil para limpiar la carpeta de trabajo antes de realizar una operación importante, como por ejemplo hacer un commit o crear una nueva rama.

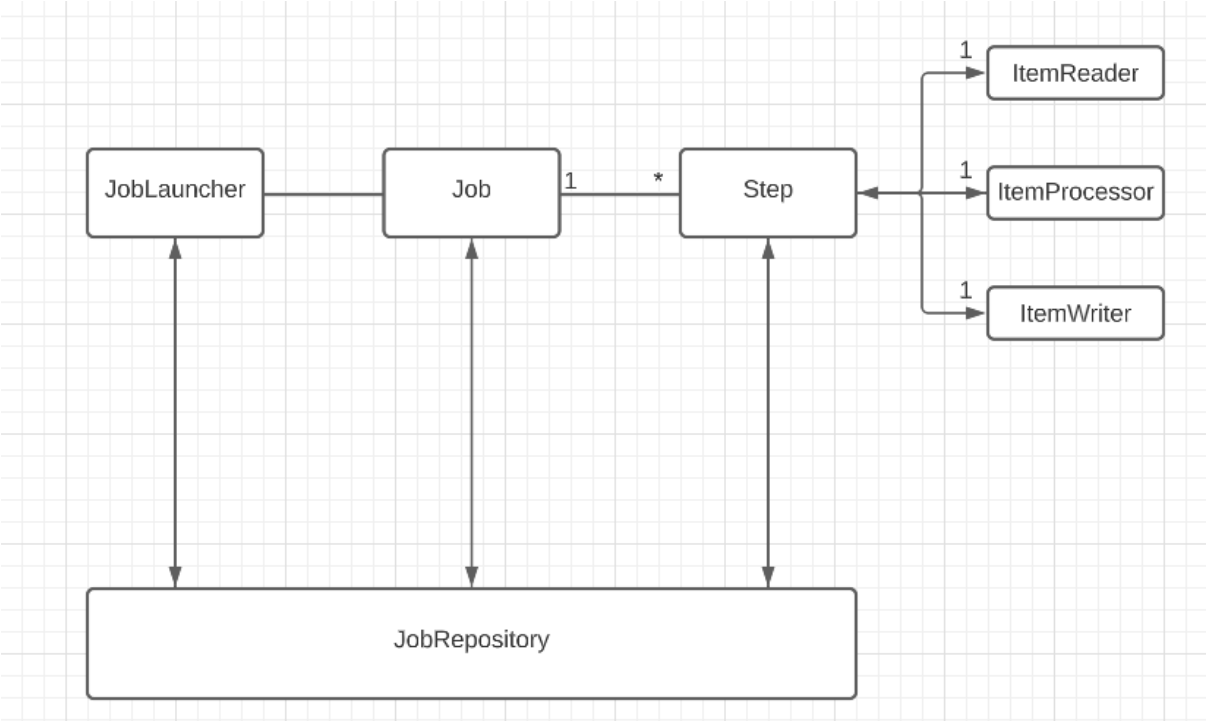
Cherry-pick: Comando de Git que permite seleccionar y aplicar cambios individuales de una rama a otra. Este comando es útil cuando se necesita aplicar cambios específicos de una rama a otra sin necesidad de fusionar las ramas completamente.

Pasos para usar cherry-pick:

1. Asegúrate de que estás en la rama donde quieres aplicar el cherry pick
2. Ejecuta el comando "git cherry-pick <commit-hash>" para aplicar el commit específico que quieres.
3. Si hay conflictos durante el cherry-pick, debes resolverlos manualmente.
4. Una vez que los conflictos estén resueltos, haz un "git cherry-pick --continue" para finalizar la operación de cherry-pick.
5. Finalmente, haz un "git push" para enviar los cambios a tu repositorio remoto.

Spring batch

Spring Batch es un framework de procesamiento por lotes en Java que se utiliza para automatizar tareas repetitivas y que pueden ser largas y complejas. Con Spring Batch, puedes procesar grandes cantidades de datos en un formato uniforme, lo que aumenta la eficiencia y la velocidad.



Job repository: Es un almacén de datos que mantiene información sobre los trabajos y sus estados.

Job Launcher: Es un componente que se encarga de iniciar los trabajos.

Job: Es una unidad de procesamiento por lotes que consta de una o más Steps.

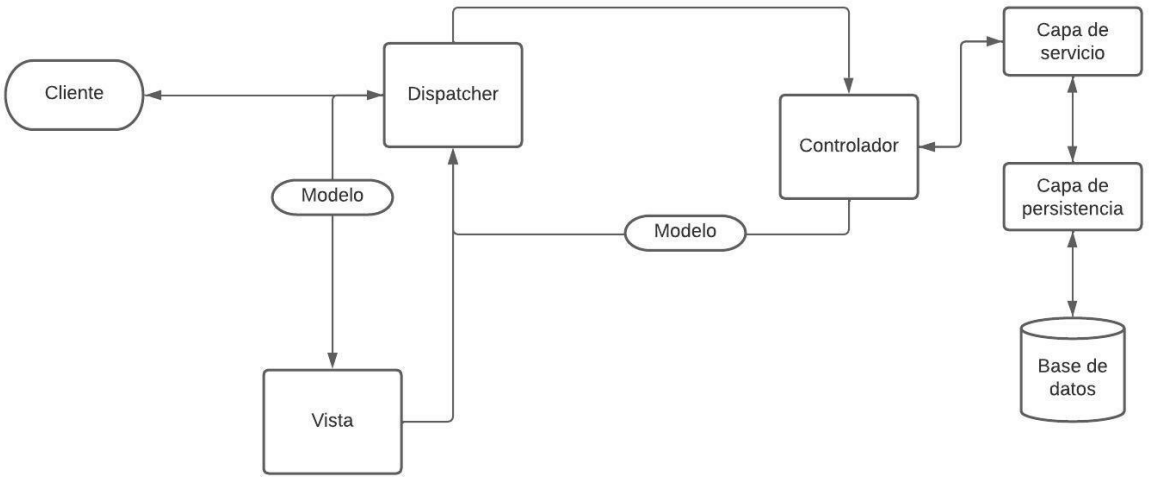
Step: Es una unidad de procesamiento que contiene una secuencia definida de acciones que se realizan sobre un lote de datos.

Item Reader: Es un componente que lee los datos de un origen específico, como un archivo o una base de datos, y los proporciona al procesador.

Item Writer: Es un componente que escribe los datos resultantes a un destino específico, como un archivo o una base de datos.

Item Processor: Es un componente que se encarga de realizar la transformación de los datos leídos por el Item Reader antes de ser escritos por el Item Writer.

Spring MVC



En la arquitectura MVC de Spring, los controladores son clases Java que manejan solicitudes HTTP y proporcionan una respuesta a la vista. Cuando un usuario envía una solicitud a la aplicación, el Dispatcher la recibe y la envía a un controlador adecuado para su procesamiento. El controlador puede obtener y modificar los datos del modelo y luego seleccionar la vista apropiada para mostrar la respuesta.