

Preguntas de la semana 3

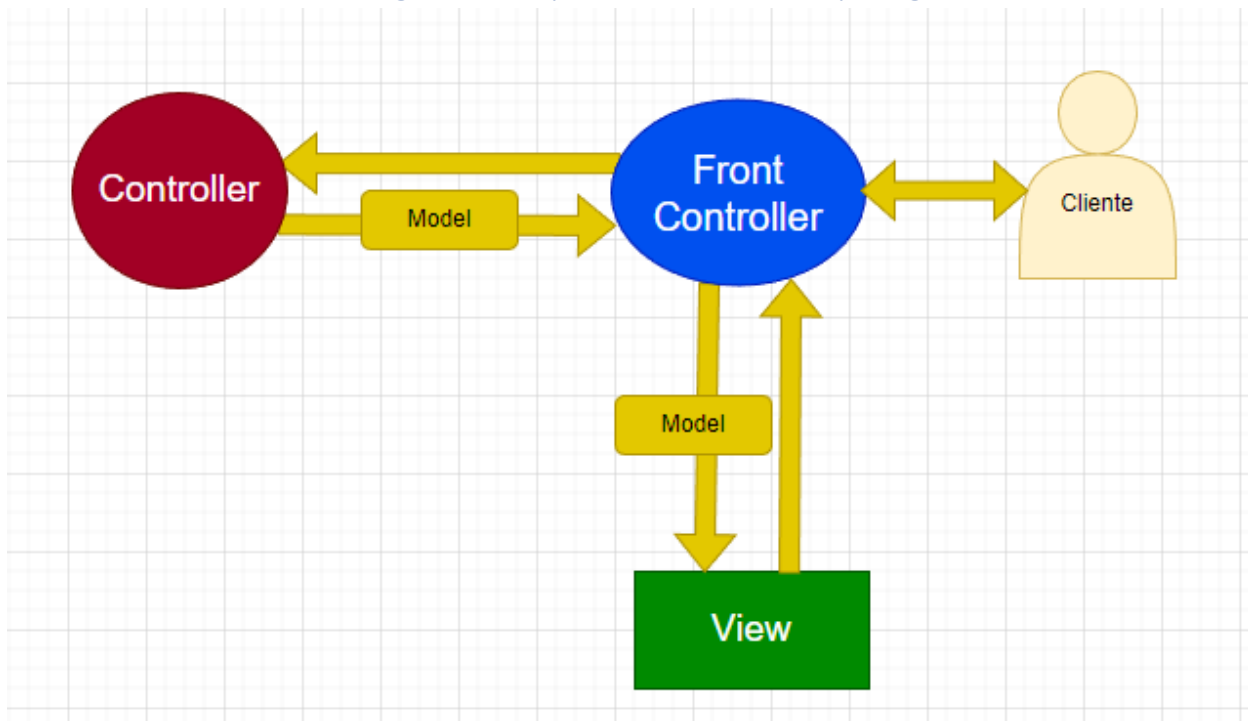
---

Clever Armando León Valdez

## Indice

Indice.....	2
Diagrama el patrón MVC de spring.....	3
Comandos git .....	4
Pull request .....	4
Fork .....	4
Rebase.....	4
Stach.....	5
Clean .....	5
Cherry-pick.....	5
Explicar y diagramar spring Bach .....	6

## Diagrama el patrón MVC de spring



Funcionamiento del MVC en spring:

- Todas las peticiones HTTP se canalizan a través del *front controller*.
- El *front controller* averigua a partir del url a qué Controller hay que llamar para servir la petición. Para esto se usa un `HandlerMapping`.
- Se llama al Controller, que ejecuta la lógica de negocio, obtiene los resultados y los devuelve al servlet, encapsulados en un objeto del tipo `Model`. Además, se devolverá el nombre lógico de la vista a mostrar normalmente devolviendo un `String`
- Un `ViewResolver` se encarga de averiguar el nombre físico de la vista que se corresponde con el nombre lógico del paso anterior.
- Finalmente, el *front controller* redirige la petición hacia la vista, que muestra los resultados de la operación realizada.

## Comandos git

### Pull request

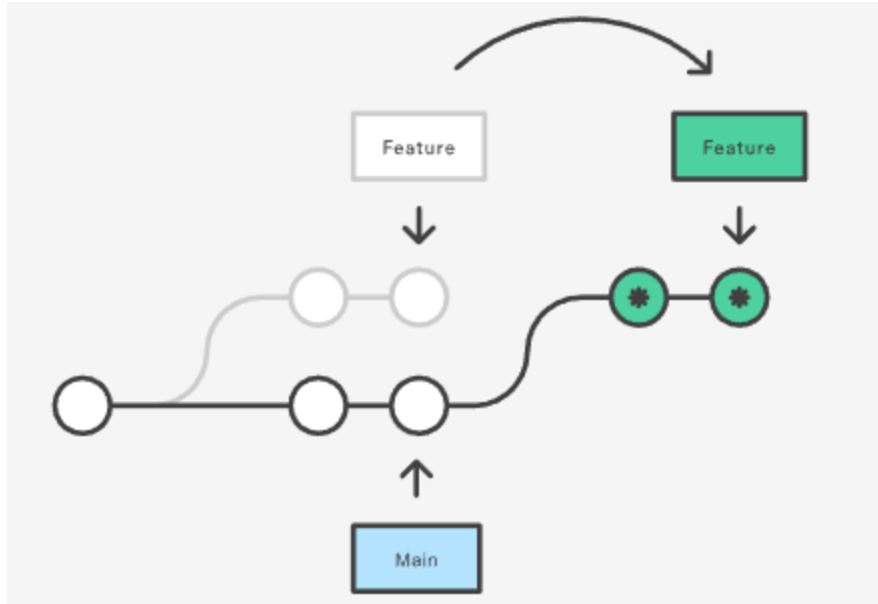
- Los pull requests son la forma de contribuir a un proyecto grupal o de código abierto.
- Se realiza desde la pagina de git del proyecto
- Como se realiza
  - o Se trabaja en una rama paralela los cambios que se desean git checkout -b <rama>.
  - o Se hace un commit a la rama git commit -am '<Comentario>'.
  - o Se suben al remoto los cambios git push origin <rama>.
  - o En GitHub se hace el pull request comparando la rama master con la rama del fix.
  - o Uno, o varios colaboradores revisan que el código sea correcto y dan feedback (en el chat del pull request).
  - o El colaborador hace los cambios que desea en la rama y lo vuelve a subir al remoto (automáticamente jala la historia de los cambios que se hagan en la rama, en remoto).
  - o Se aceptan los cambios en GitHub.
  - o Se hace merge a master desde GitHub.

### Fork

- Básicamente sirve para crear una copia de un repositorio en tu cuenta de usuario
- Este fork sera independiente al original, y evolucionara de diferente manera
- Una vez hecho el fork existirán dos repositorios distintos.
- Una vez creado el fork, puedes realizar cambios y solicitar el pull request a través de la página de GitHub, que es básicamente una solicitud para que tu código se fusione con el código del repositorio donde has colaborado.

### Rebase

- es el proceso de mover o combinar una secuencia de confirmaciones a una nueva confirmación base.
- Desde una perspectiva de contenido, el rebase está cambiando la base de su rama de un commit a otro haciendo que parezca que ha creado su rama a partir de un commit diferente.
- Internamente, Git logra esto creando nuevas confirmaciones y aplicándolas a la base especificada.



### Stash

- puedes almacenar temporalmente una captura de tus cambios sin enviarlos al repositorio.
- Uso
  - o git stash save "mensaje opcional para ti" - Para guardar tus cambios en el stash
  - o git stash list - Para ver lo que hay en tu stash
  - o git stash apply NOMBRE-DEL-STASH - aplica los cambios y deja una copia en el stash
  - o git stash pop NOMBRE-DEL-STASH - aplica los cambios y elimina los archivos del stash
  - o git stash drop NOMBRE-DEL-STASH – eliminar stash

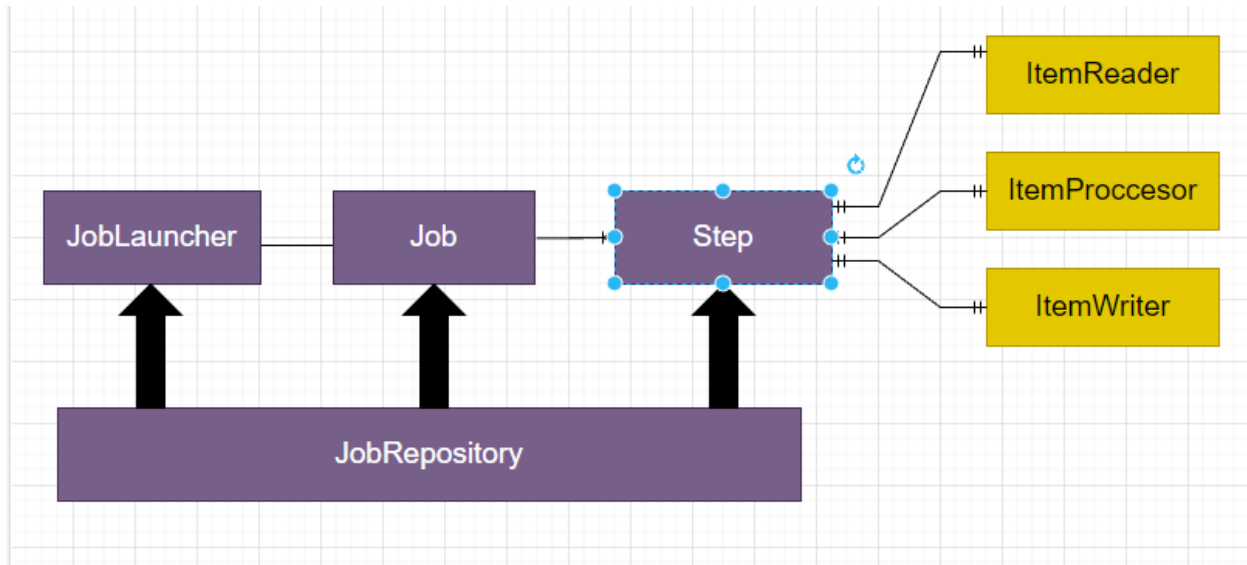
### Clean

- El comando se usa para limpiar los archivos no rastreados en el repositorio.
- Uso del comando
  - o git clean -n: para correr en seco.
  - o git clean -f: eliminación forzada de archivos.
  - o git clean -f -x: eliminar archivos gitignore
  - o git clean -f -d: elimine los directorios no rastreados.

### Cherry-pick

- Permite tomar uno o varios commits de una rama, sin hacer un merge completo.
- Se usa principalmente si no desea fusionar toda la rama y desea algunas de las confirmaciones.

## Explicar y diagramar spring Batch



Observamos diferentes elementos:

- JobRepository: es el componente encargado de la persistencia de metadatos relativos a los procesos tales como procesos en curso o estados de las ejecuciones.
- JobLauncher: 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.
- Step: Un step es un elemento independiente dentro de un Job (un proceso) que representa una de las fases de las que está compuesto dicho proceso. Un proceso Job debe tener al menos un step.

Aunque no es obligatorio, un step puede estar compuesto de tres elementos

- ItemReader: Elemento responsable de leer datos de una fuente de datos como BBDD, fichero, cola de mensajes, etc.
- 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.