

Git y personas



Febrero 2019

Jesús López
jesuslc.com

Sponsors

**MARBELLA.
DIGITAL**



europa



¿Quién soy?

Jesús López

Tech Lead en Qashops (Vente Privée Company)

Veepee 

Ingeniero en Informática

Professional Scrum Master Certified

jesuslc.com

@jeslopcru

github.com/jeslopcru

¿Quién utiliza git?

¿De qué vamos a hablar?

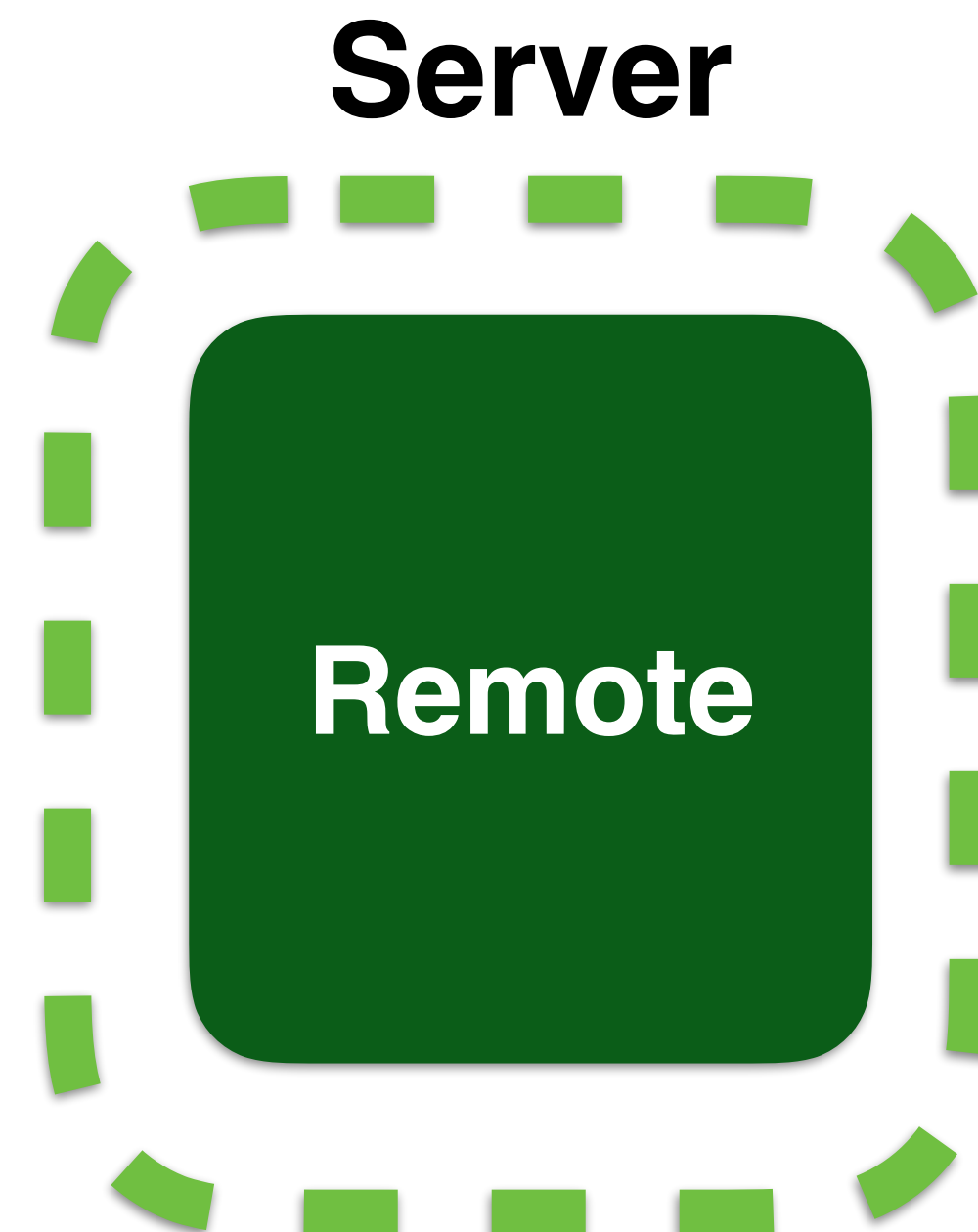
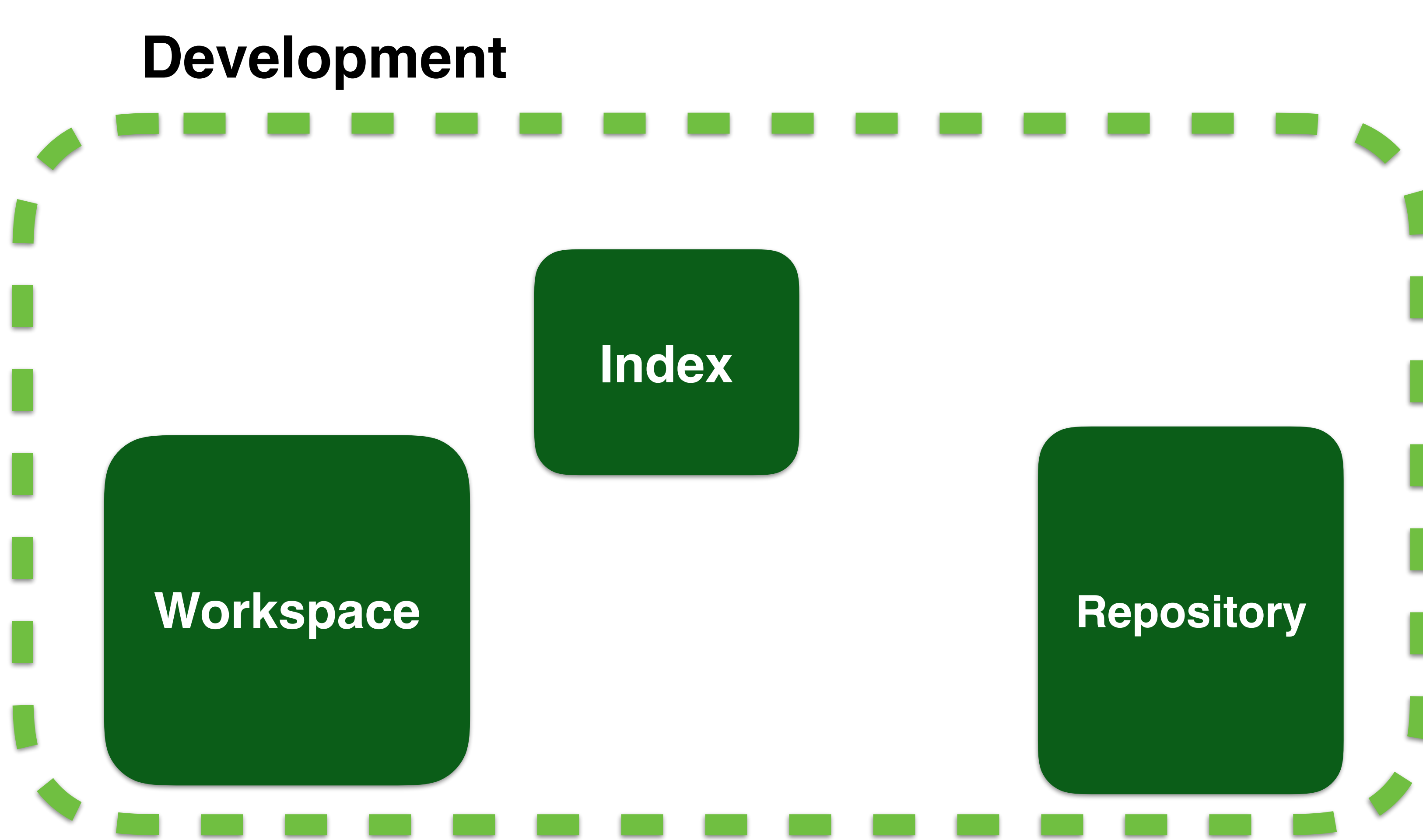
- ¿Qué es git?
- Trabajando juntos
- Lecciones aprendidas

¿Qué es git?

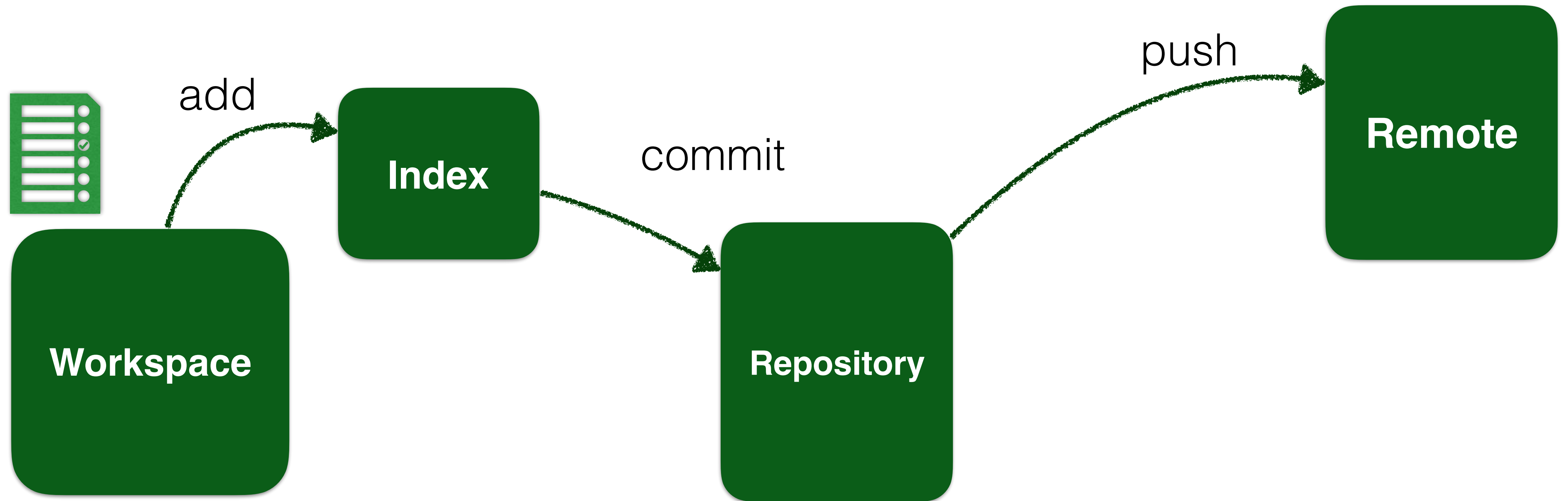
¿Qué es git?

Es un software de **control de versiones** diseñado por Linus Torvalds, que te ayuda a llevar el registro de los cambios en el código fuente.

¿Qué es git?



¿Qué es git?

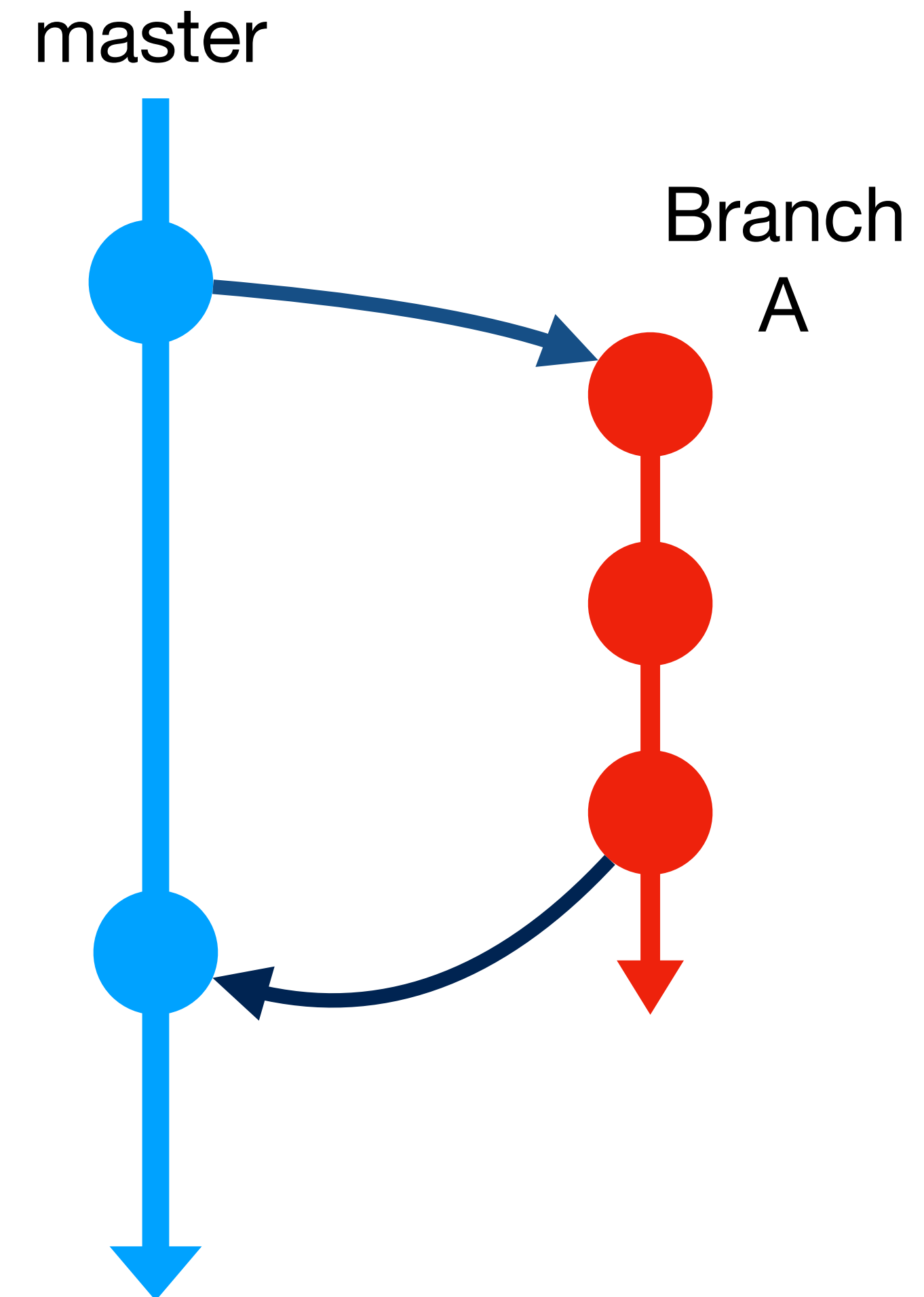


5 comandos básicos de git

- git checkout
- git commit
- git pull
- git merge
- git push

5 comandos básicos de git

- git checkout: **cambiar de rama**
- git commit: **crear una bolita**
- git pull: **traer cambios del repositorio**
- git merge: **fusionar**
- git push: **enviar cambios al repositorio**



5 comandos básicos de git y las bases para trabajar en equipo

- git checkout
 - git commit
 - git pull
 - git merge
 - git push
- **Comunicación**
 - **Confianza**
 - **Simplicidad**

OBJETIVO

Aportar valor al negocio a
través del código
minimizando el riesgo

Trabajando juntos

Trabajando juntos

Tareas

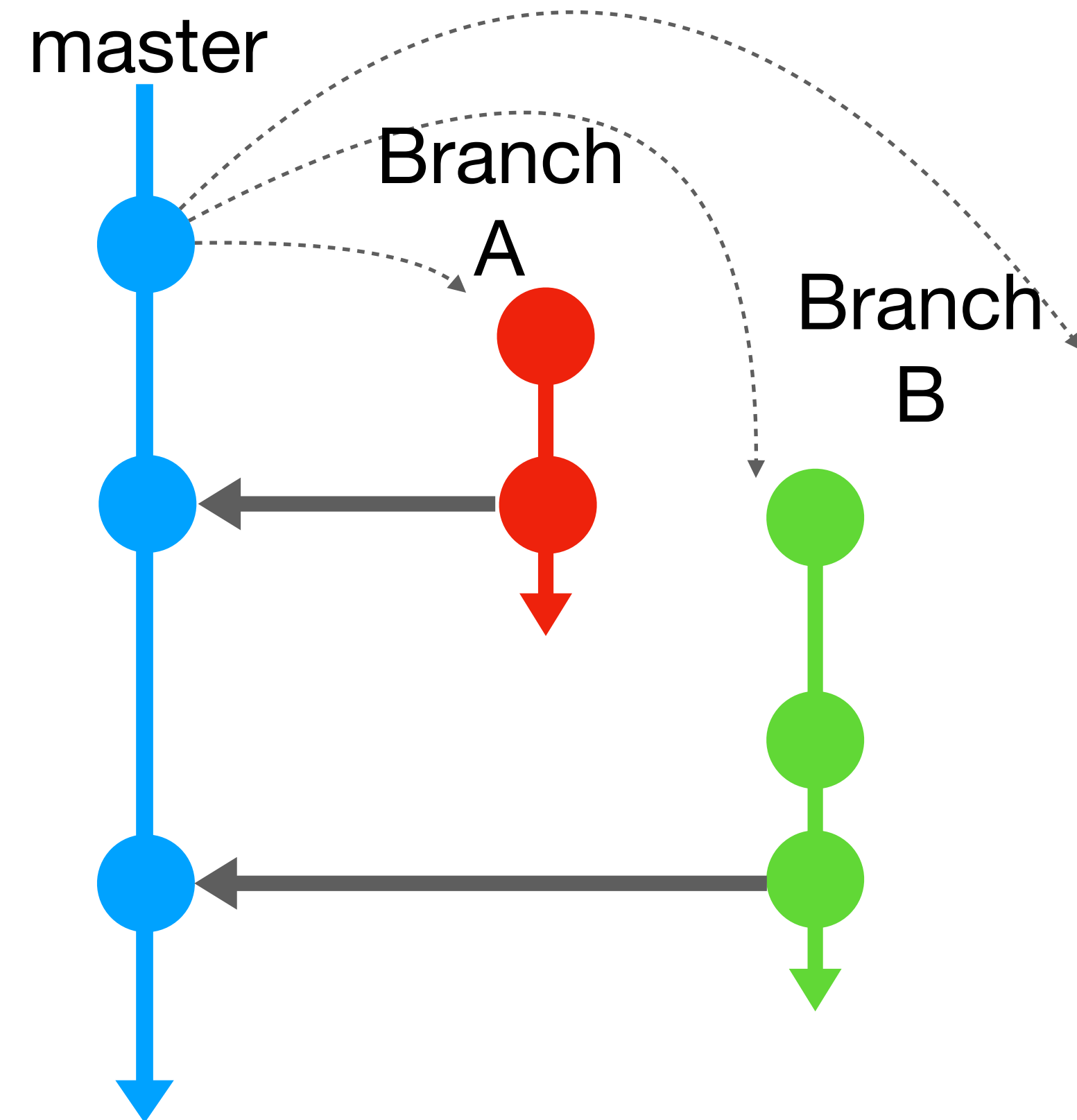


Es más fácil **colaborar** si todo está en el mismo sitio

Trabajando juntos

Ramas

Nombre de la rama el ID-Tracker



Trabajando juntos

Commits

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Trabajando juntos

Commits

ID-Tracker: Breve resumen de los cambios realizados

Cuerpo: Explicación opcional más detallada que aporta contexto e información adicional.

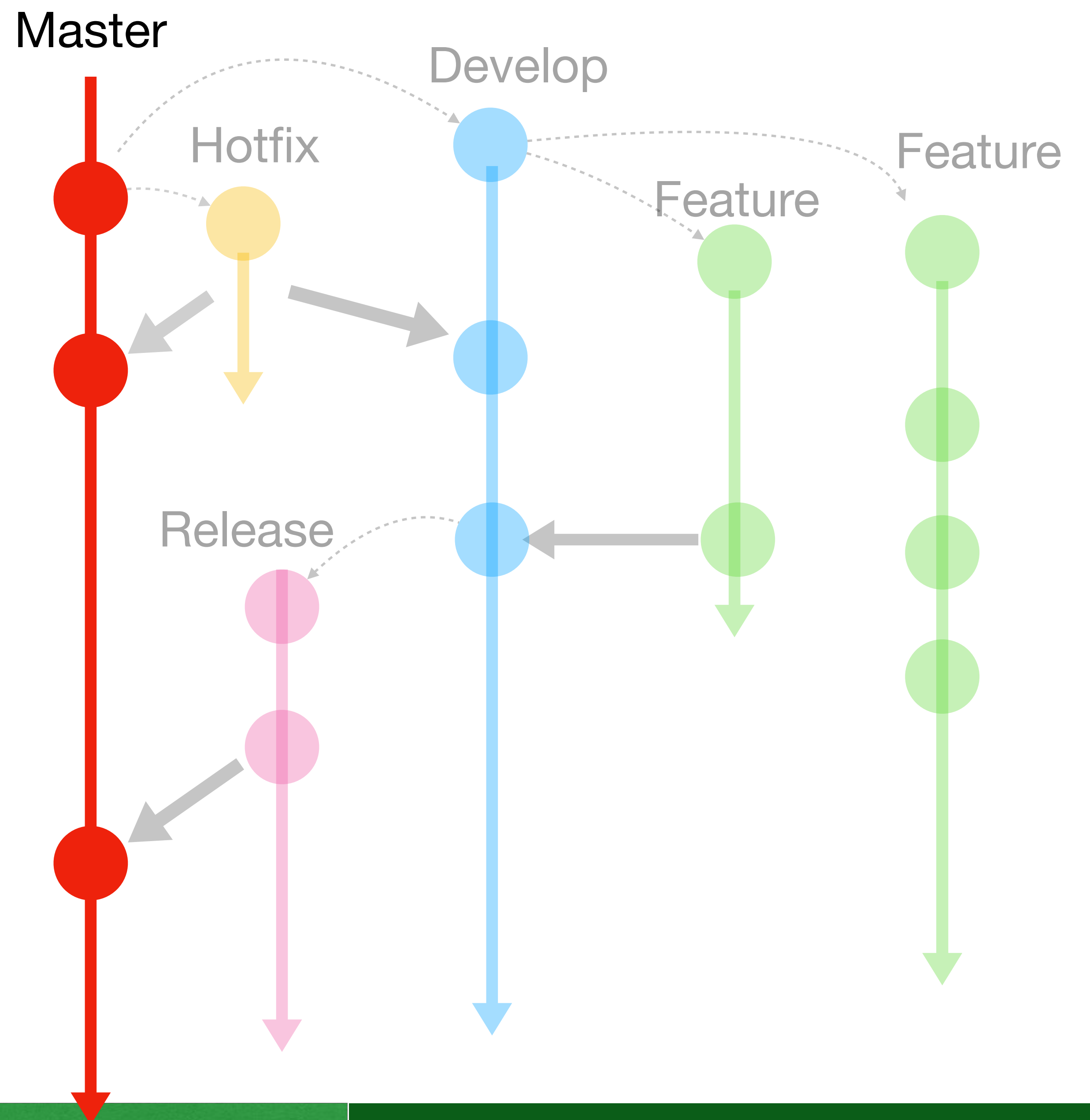
**No
blame**

Estrategias de branching

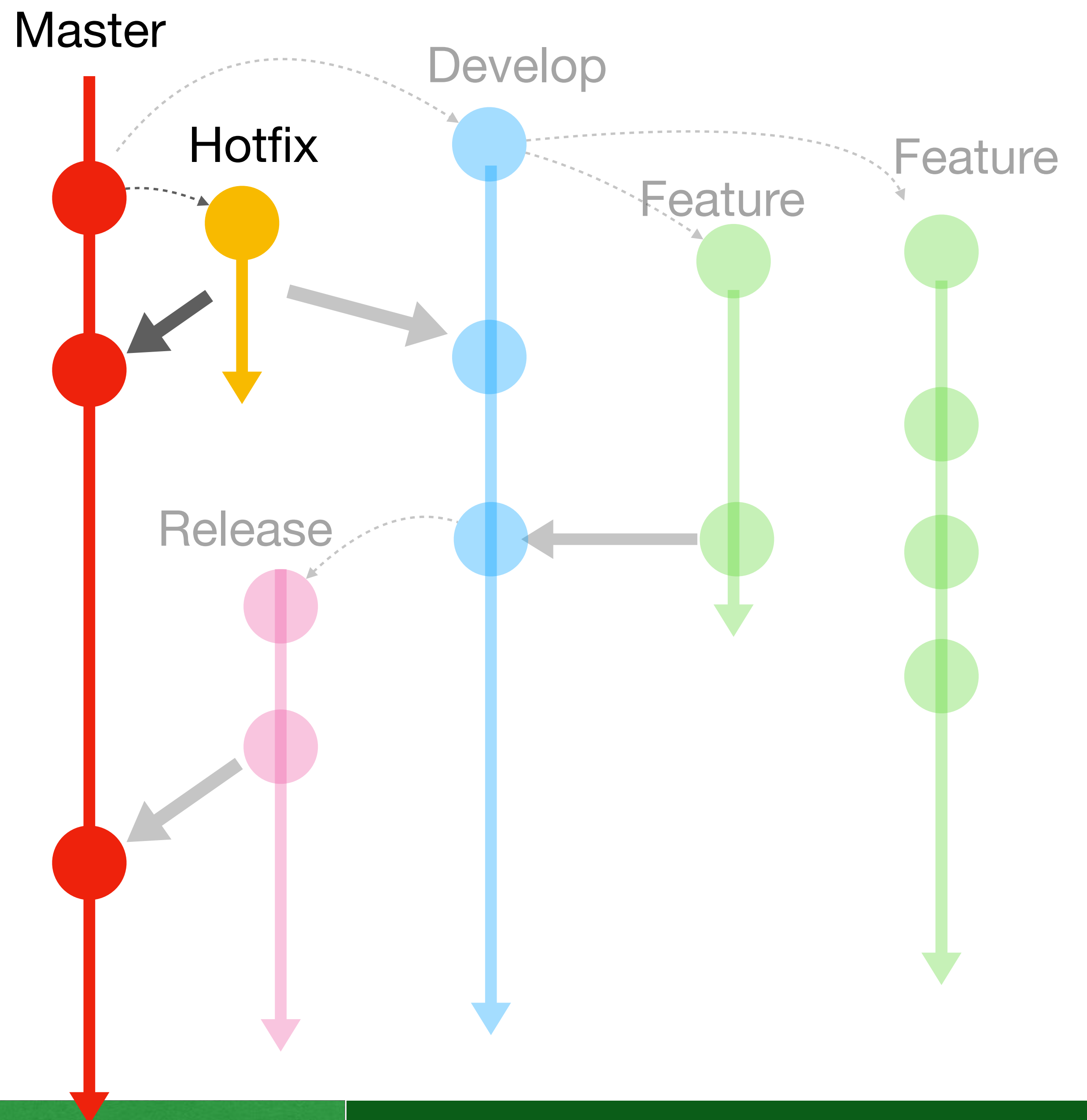
Git flow

El que todo el mundo conoce

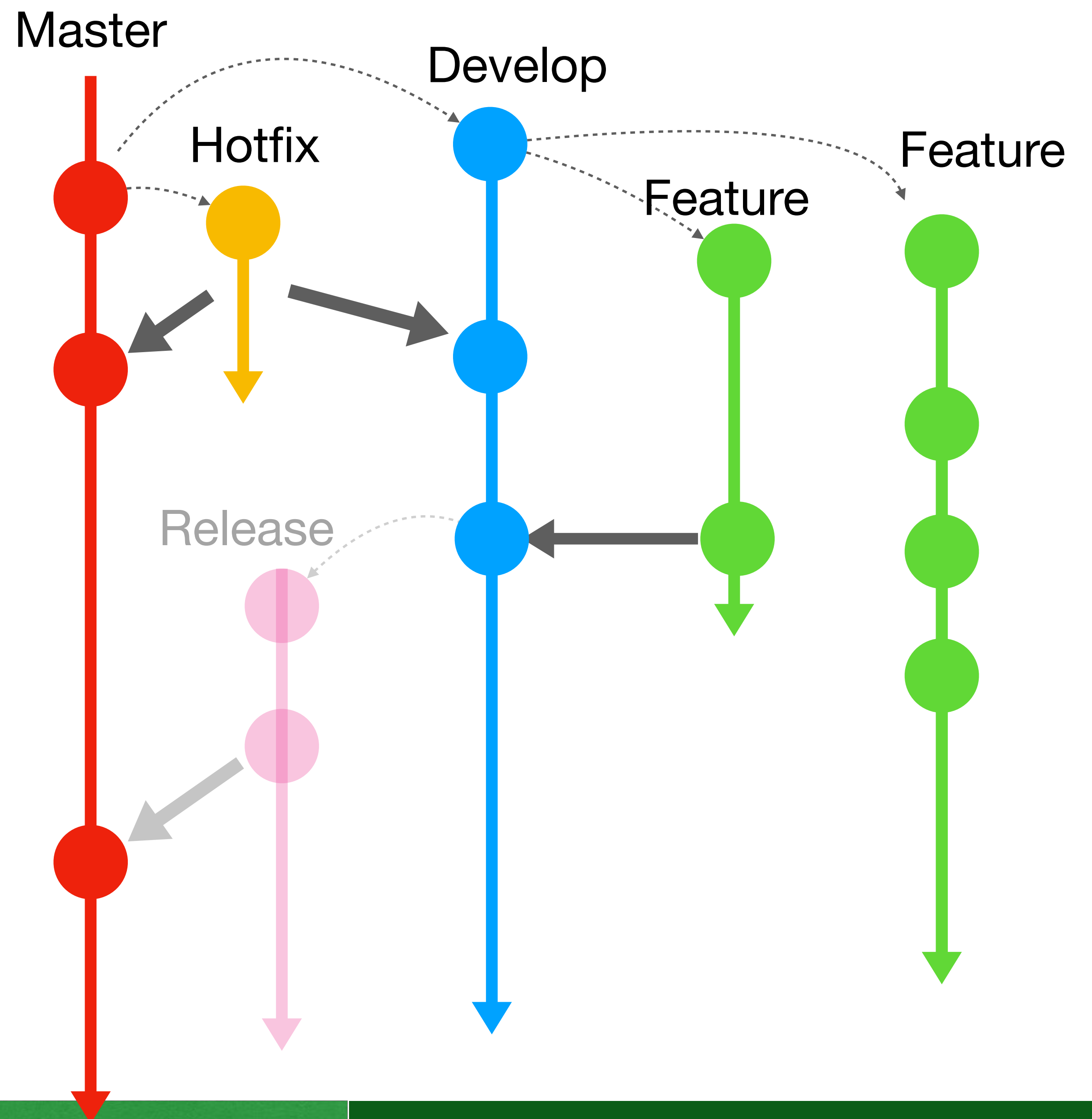
Git flow



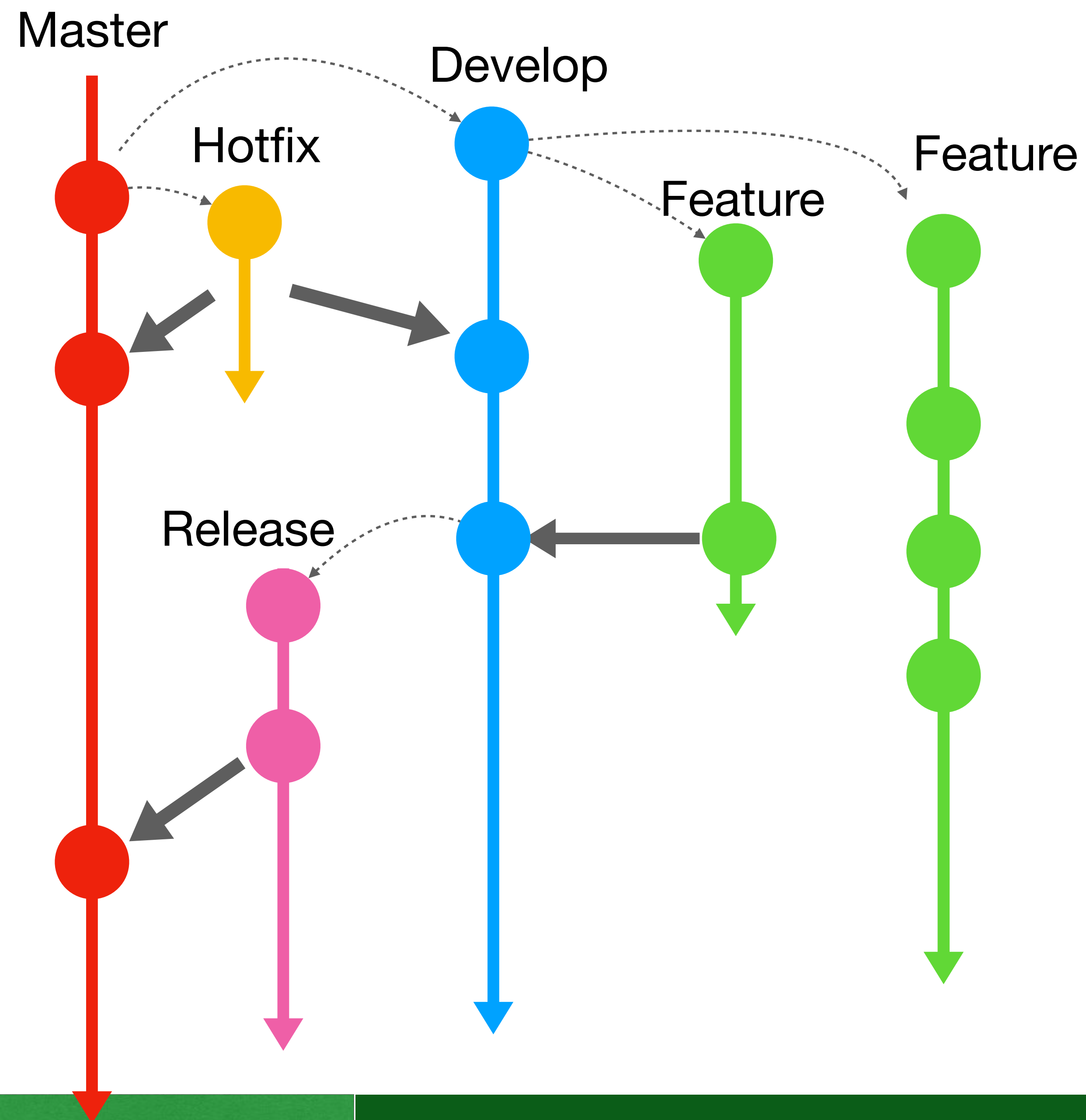
Git flow



Git flow

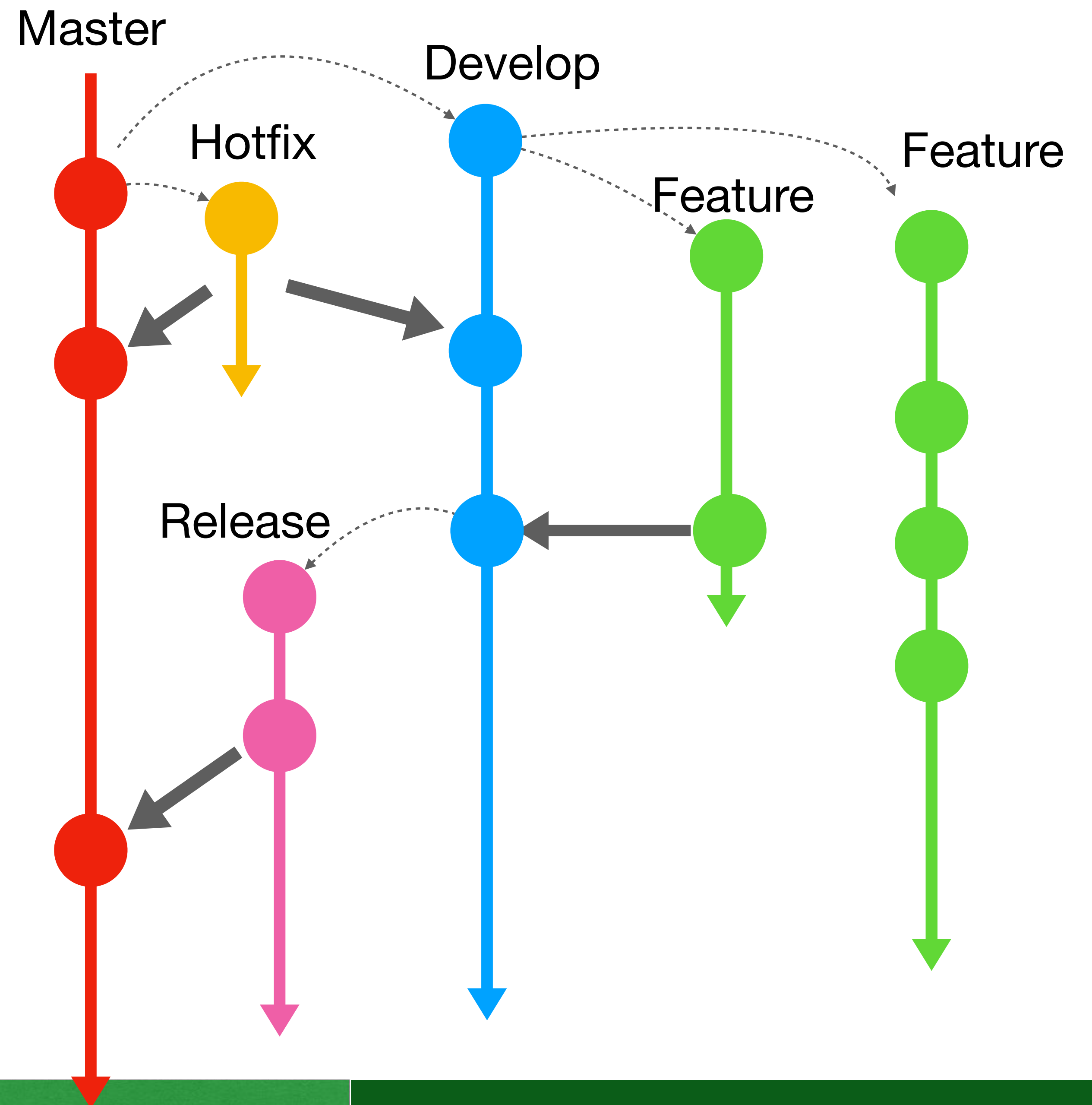


Git flow



Git flow

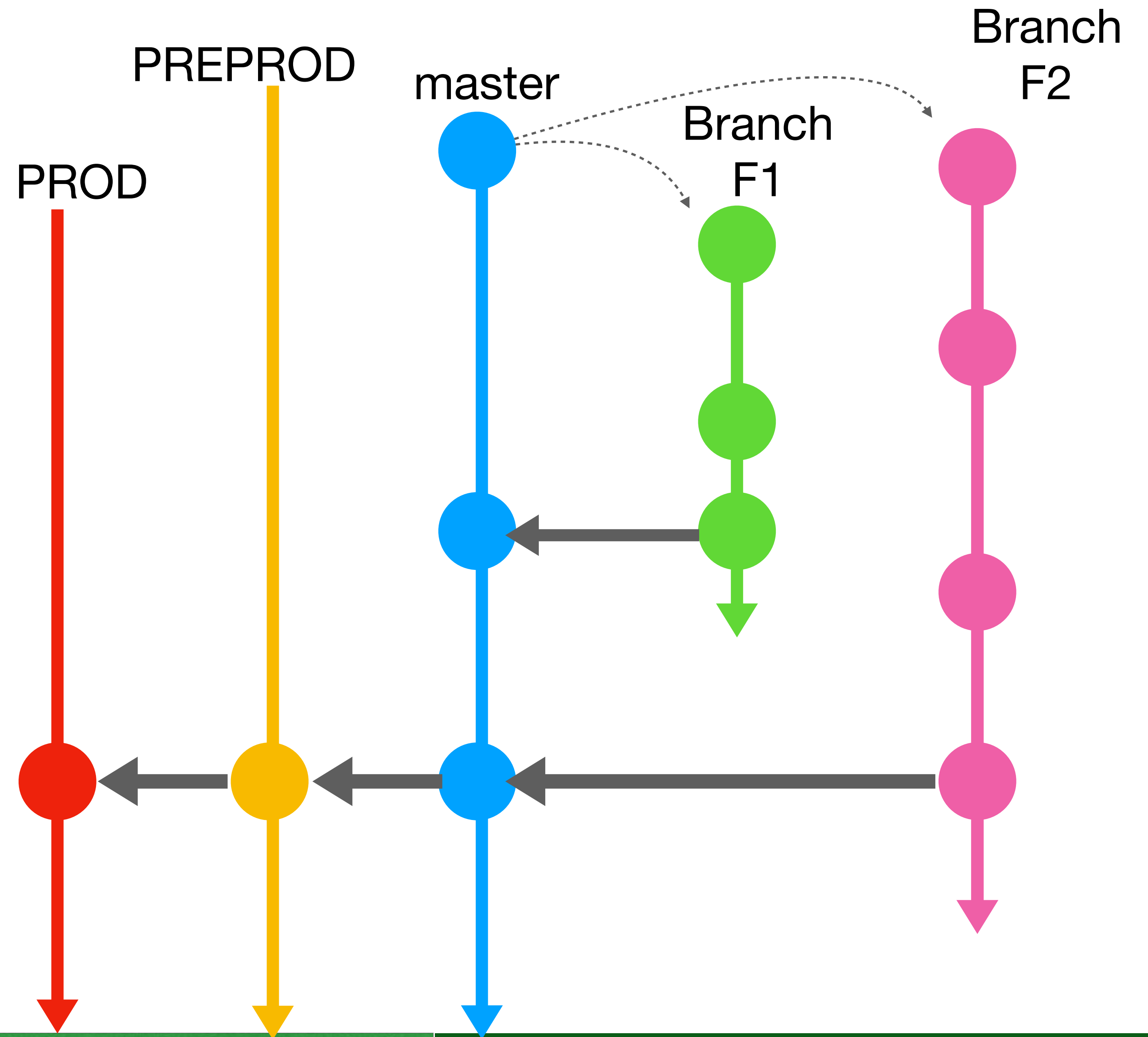
- Mucha documentación y herramientas
- Complejo para equipos pequeños
- Todo está ordenado



Environment based

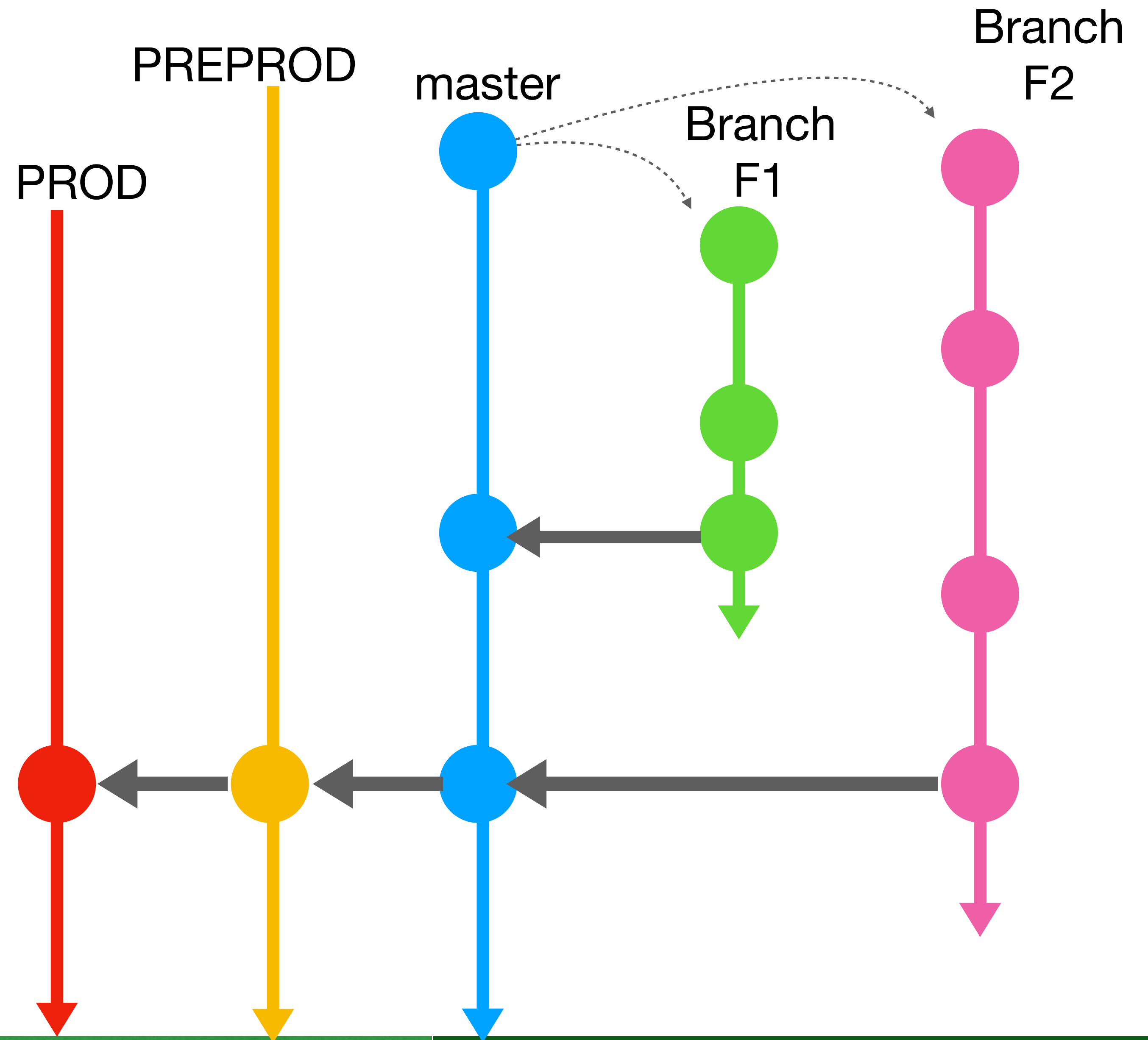
Revisar antes de subir

Environment based



Environment based

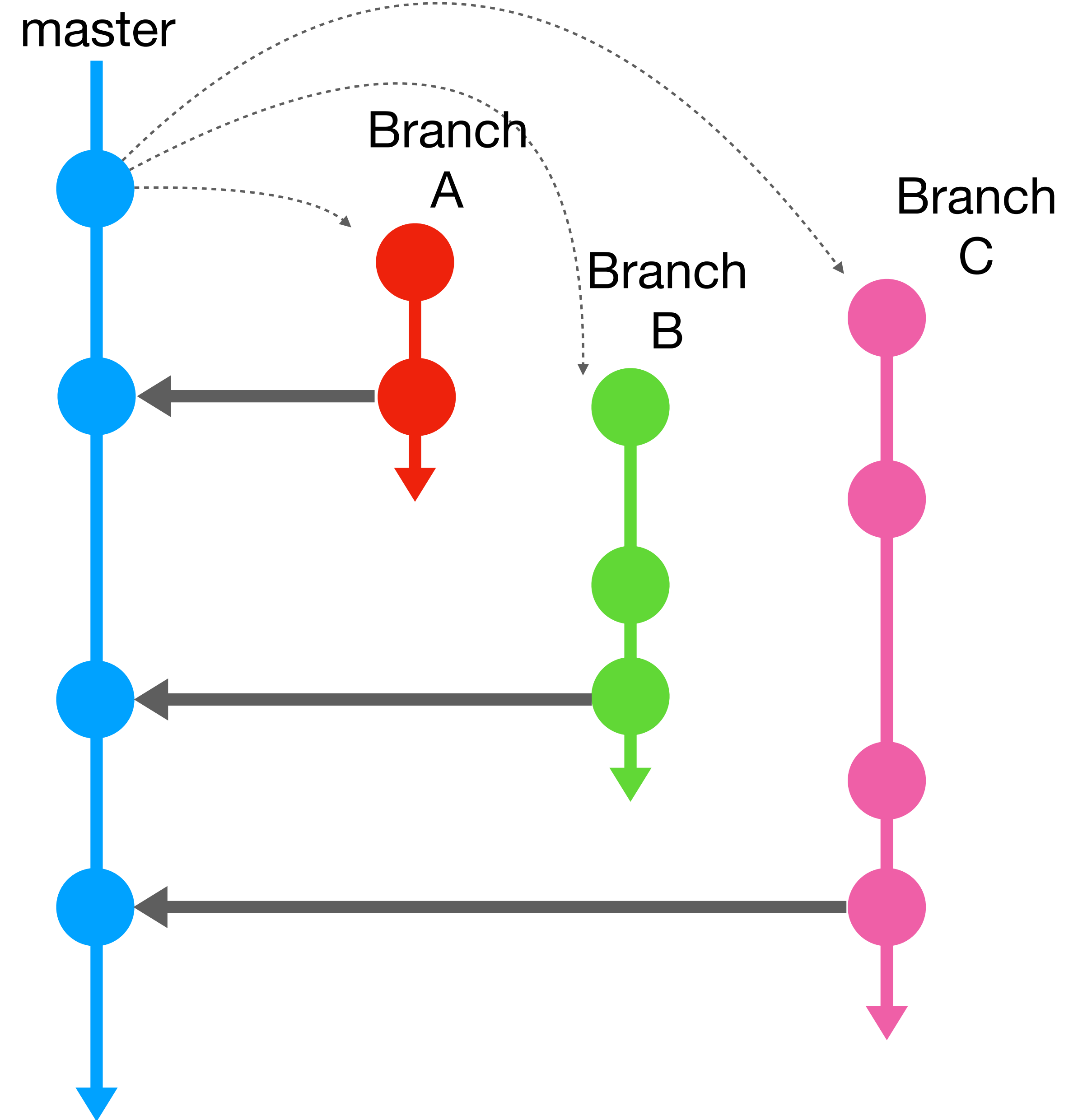
- Probar antes de subir en un entorno parecido al real
- Validación
- Es necesario automatizar



Mainlain branch

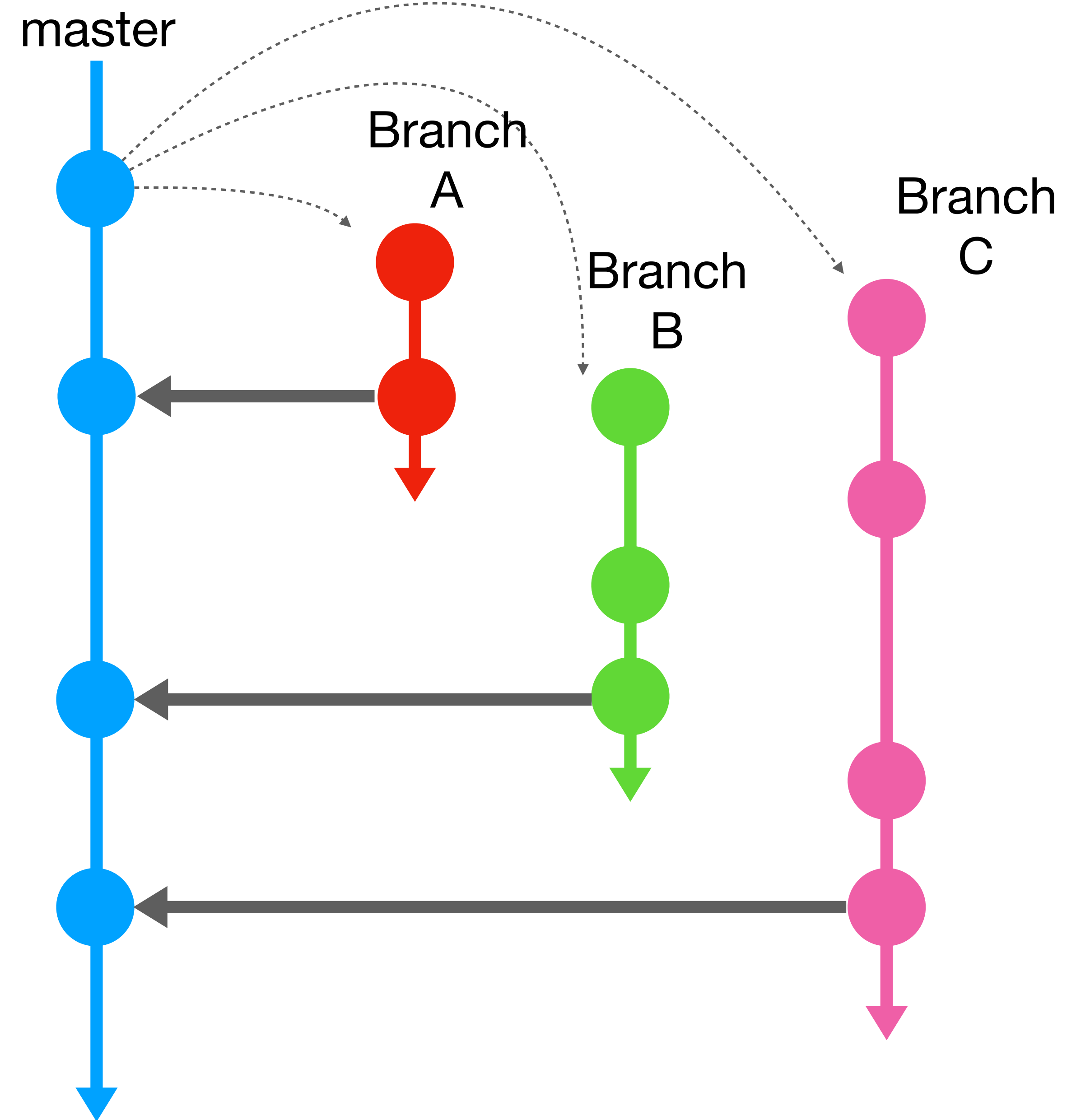
Simplicidad

Mainlain branch



Mainlain branch

- Equipo pequeño
- Code review + Tests + revisión
- Definición de historias de usuario

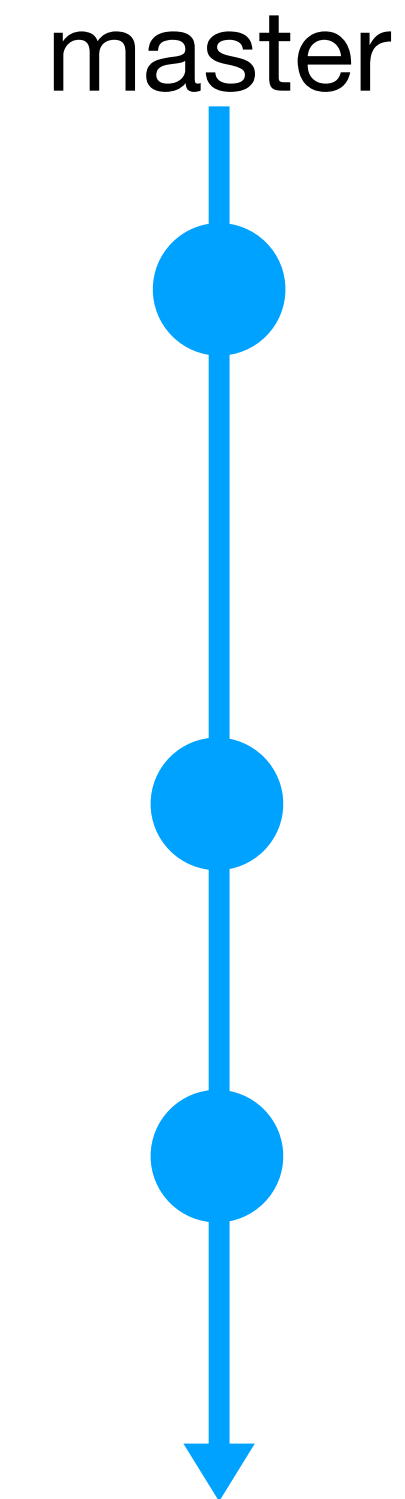


Trunk based

Branches are for monkeys

Trunk based

- Push directo a master
- Testing + automatización
- **Lo importante no es hacer esta estrategia sino el camino recorrido para usarla**



Lecciones aprendidas

Lecciones aprendidas

Estrategia de branching

- * Equipo
- * ¿Aplicación web? ¿iOS? ¿Android?
- * ¿Validación/QA? ¿Historias de Usuario?
- * Automatización: test + despliegue

Estrategias branching

Git flow	Mucha documentación	Sencillo de entender	Más ramas que personas
Environment based	Producción es “especial”	Test	Depliegue automatizado
Mainlain	Feature toggles	Sin test es arriesgado	Ramas cortas
Trunk based	Obligatorio tener test	Madurez	Todo automatizado

Lecciones aprendidas

OBJETIVO: Aportar valor al negocio a través del código minimizando el riesgo

Lecciones aprendidas

- No hay una mejor que otra
- Master **siempre** debe ser “releseable”
- Cuidado con las “ramas largas”
- Comunicación, personas y confianza
- Automatizar: test + despliegue

Lecciones aprendidas

- No hay una mejor que otra
- Master **siempre** debe ser “releasable”
- Cuidado con las “ramas largas”
- ¿Qué pasa si se se hace rollback?
- Flujo simple
- Comunicación, personas y confianza
- Automatizar test + despliegue

Gracias

- <http://www.javacodegeeks.com/2015/11/git-branching>
- <https://gist.github.com/jbenet/ee6c9ac48068889b0912>
- <http://dymitruk.com/blog/2012/02/05/branch-per-feature/>
- <http://www.nomachetejuggling.com/2017/04/09/a-different-branching-strategy>
- <https://nvie.com/posts/a-successful-git-branching-model/>
- <https://trunkbaseddevelopment.com/>
- <https://about.gitlab.com/2016/07/27/the-11-rules-of-gitlab-flow/>
- <https://martinfowler.com/bliki/BranchByAbstraction.html>
- <https://rachelcarmena.github.io/2018/12/12/how-to-teach-git.html>

jesuslc.com
@jeslopcru
github.com/jeslopcru