



GITFLOW

OBJETIVOS

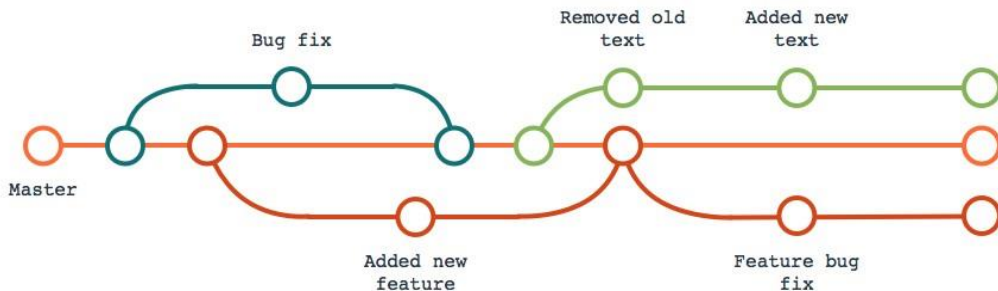
- Crear un flujo de trabajo eficiente con GIT
- Crear un modelo de confianza para evitar pérdidas de información y defectos
- Conocer el sistema de branching para mejorar la colaboración y el desarrollo colaborativo
- Conocer el concepto de Pull Request o Revisiones Cruzadas

1.

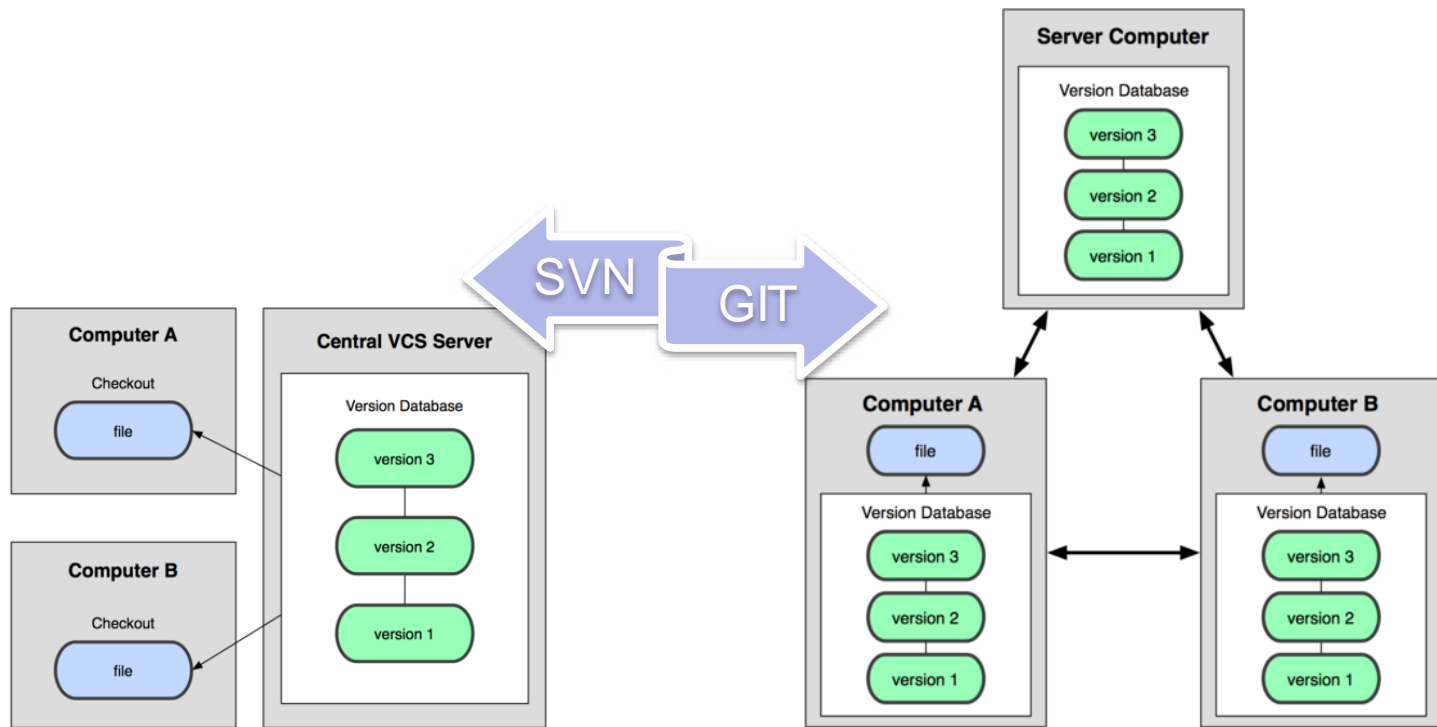
GIT

¿QUÉ ES?

Git es un **sistema de control de versiones**, que nació para dar **soporte al core Linux**, cuya principal característica es que es capaz de mantener una **gran cantidad de código fuente distribuido** y gestionado por **equipos de envergadura**, de forma **rápida y sencilla**.



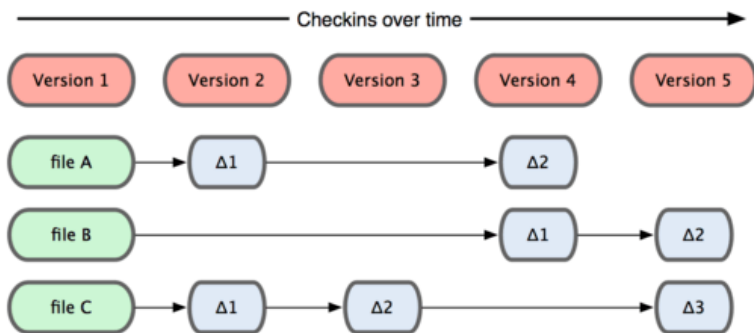
SVN vs. GIT



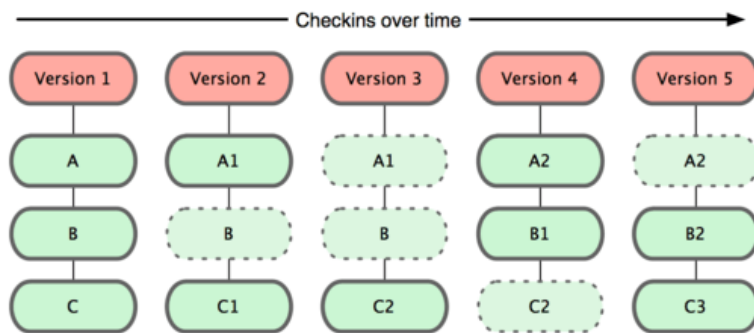
Centralizado

Distribuido

SVN vs. GIT

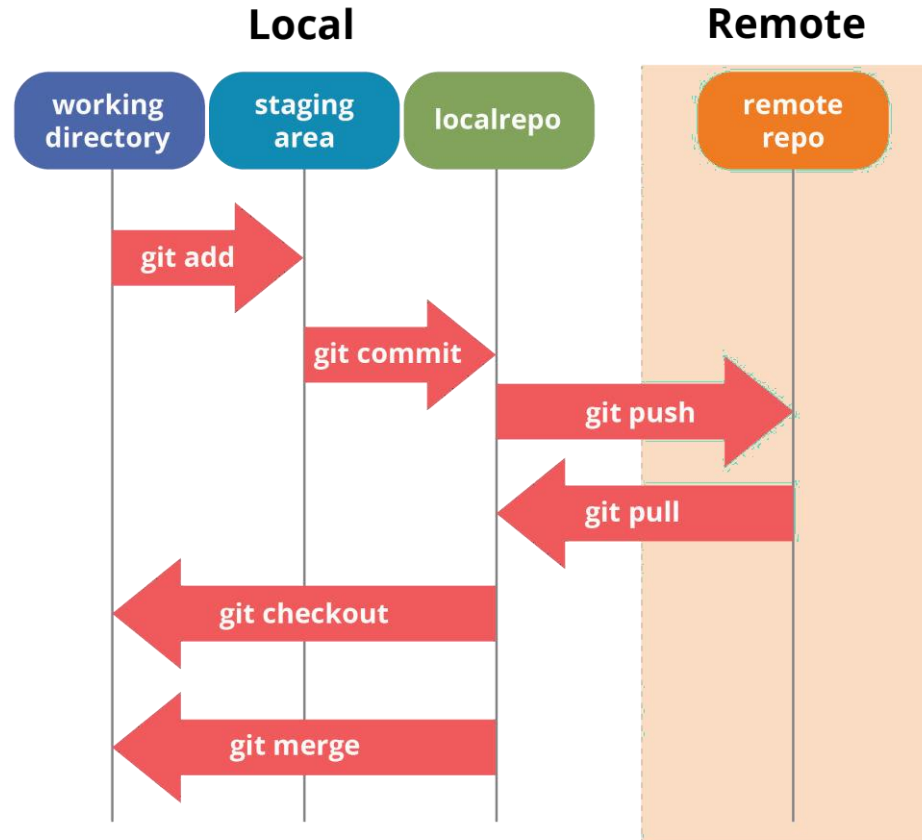


Diferencias



Instantáneas

¿CÓMO FUNCIONA?



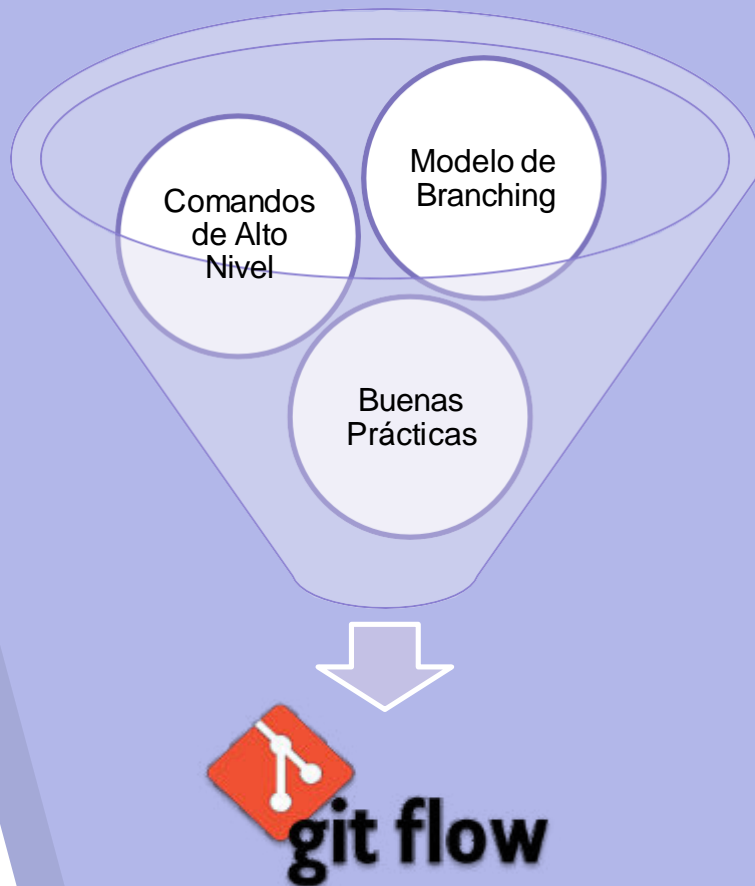


2.

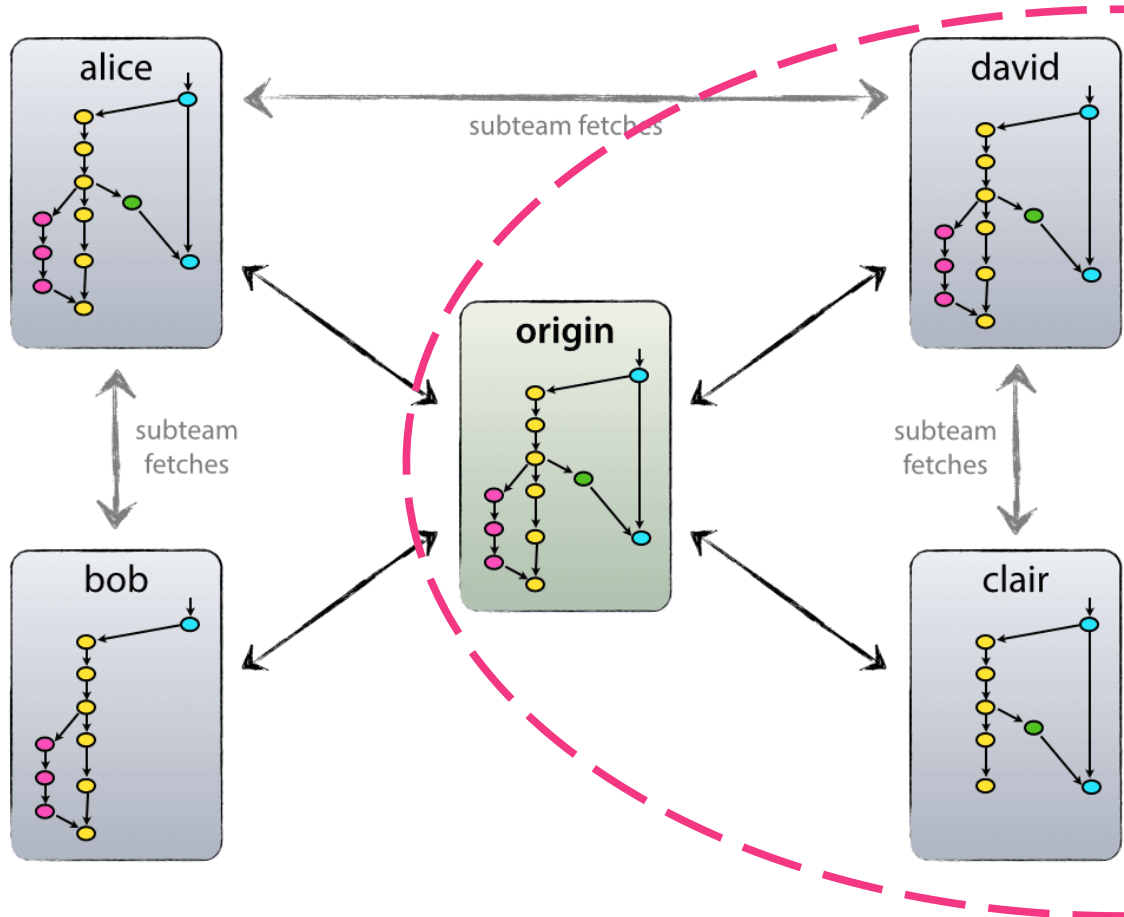
GITFLOW

¿QUÉ ES?

GitFlow es un **modelo de branching**, que propone una **estrategia de ramificación y generación de versiones** de productos usando **un repositorio Git**



DESCENTRALIZADO **PERO** CENTRALIZADO



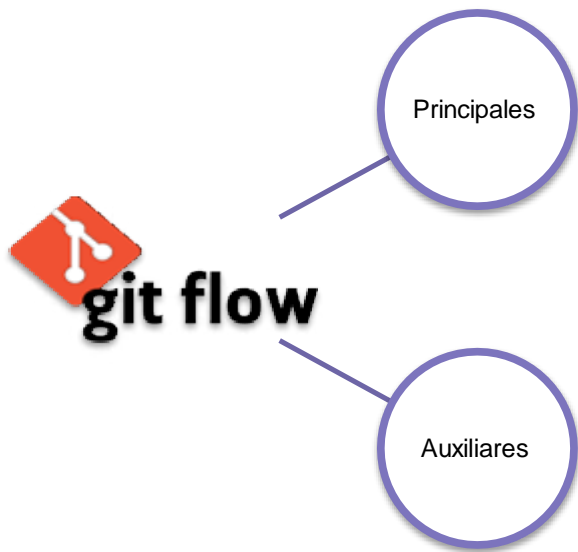
SubTeam



3.

**FLUJO DE
TRABAJO
DE GITFLOW**

DOS TIPOS DE RAMAS



- Master
- Develop

- Feature
- Release
- Hotfix

- Solo hay una de cada tipo, no hay más instancias
- Nadie puede hacer commit directamente en ellas Solo es posible incorporar código a través de otra rama
- Su nombre es igual al tipo de rama, master y develop
- Se pueden instanciar todas las veces que se considere El objetivo de instanciarlas es para trabajar directamente sobre ellas
- Su nombre está compuesto por el tipo y por su objetivo, funcionalidad, versión, etc. Por ejemplo: feature-67432-alta-usuario

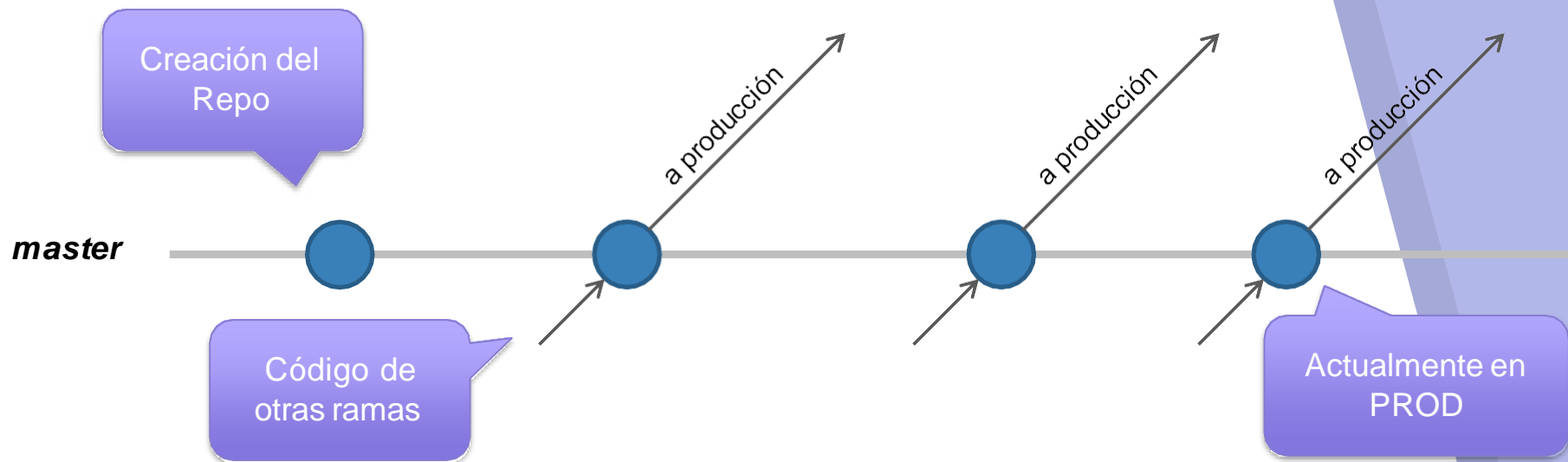
RAMA MASTER

- Es la rama principal ¡solo hay un master!

- Contiene el código que está actualmente en producción o próximo a subir

Nadie hace commit directamente sobre esta rama

- Recibe código de otras ramas a través de merges



RAMA DEVELOP

Se genera a partir
de master

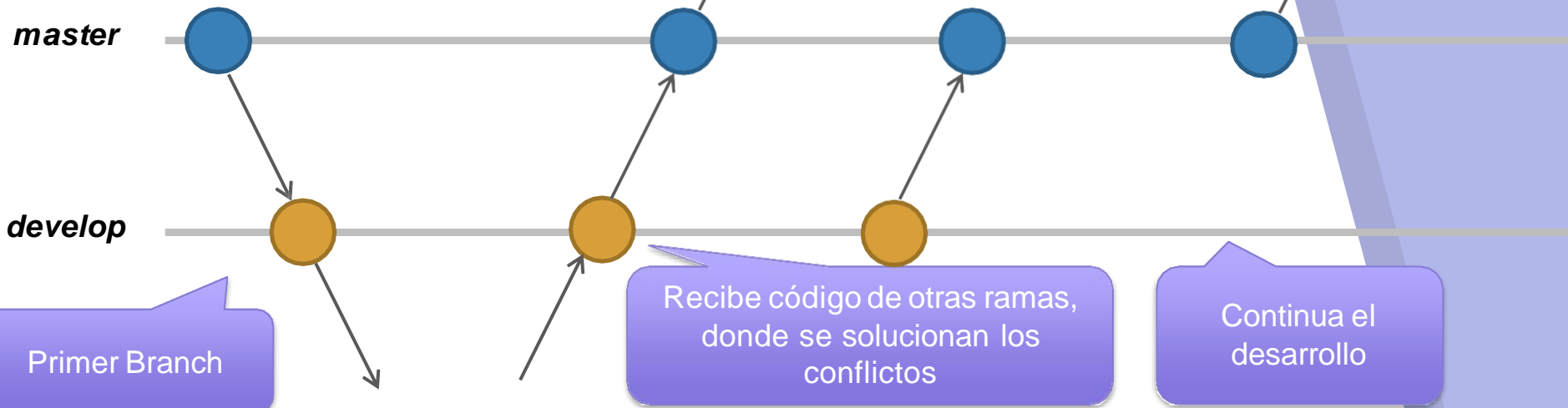
Es la otra rama principal ¡solo hay un develop!

Contiene el código
que aún no está en
producción, pero
con calidad,
completo y funcional

Nadie hace commit directamente sobre develop, o casi...

Como master, recibe
código de otras
ramas

Mantiene master
limpio y evitando
conflictos



RAMAS FEATURE

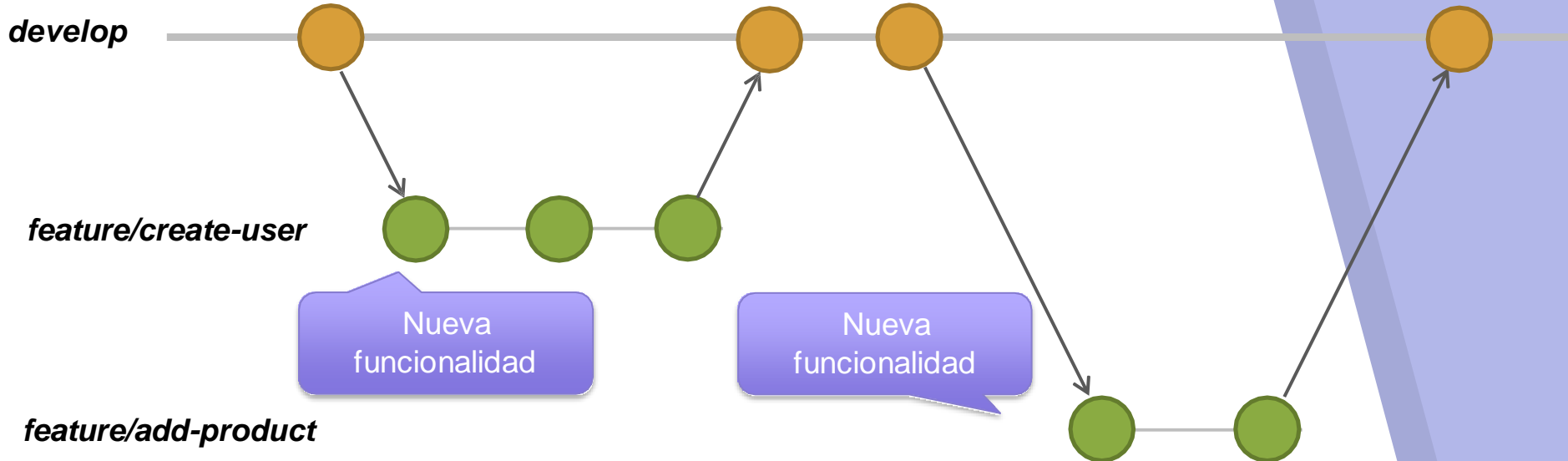
Se genera a partir de develop

Su nombre lo define el desarrollo que queramos hacer, con feature/ al comienzo

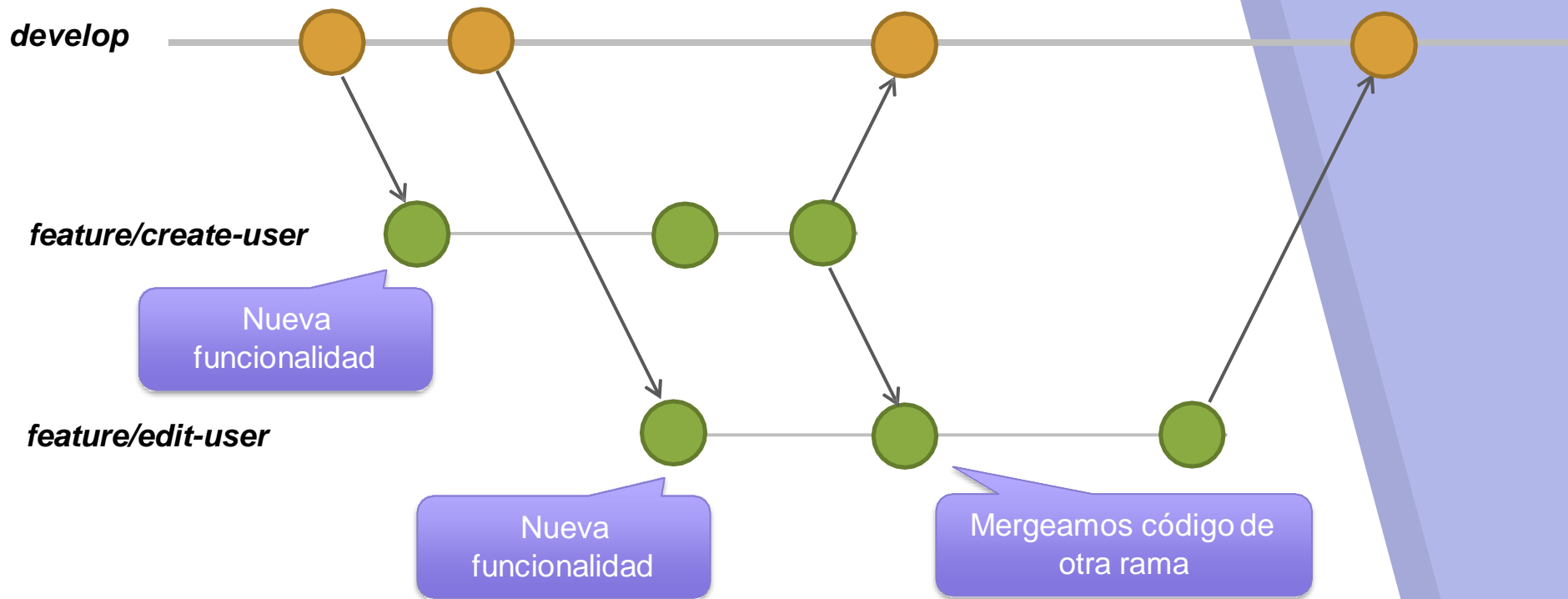
Siempre parte del último código que tengamos en develop

Contiene código vivo, que está siendo desarrollado sobre ella

Se generan para desarrollos cortos, no deben eternizarse



RAMAS FEATURE



RAMAS RELEASE

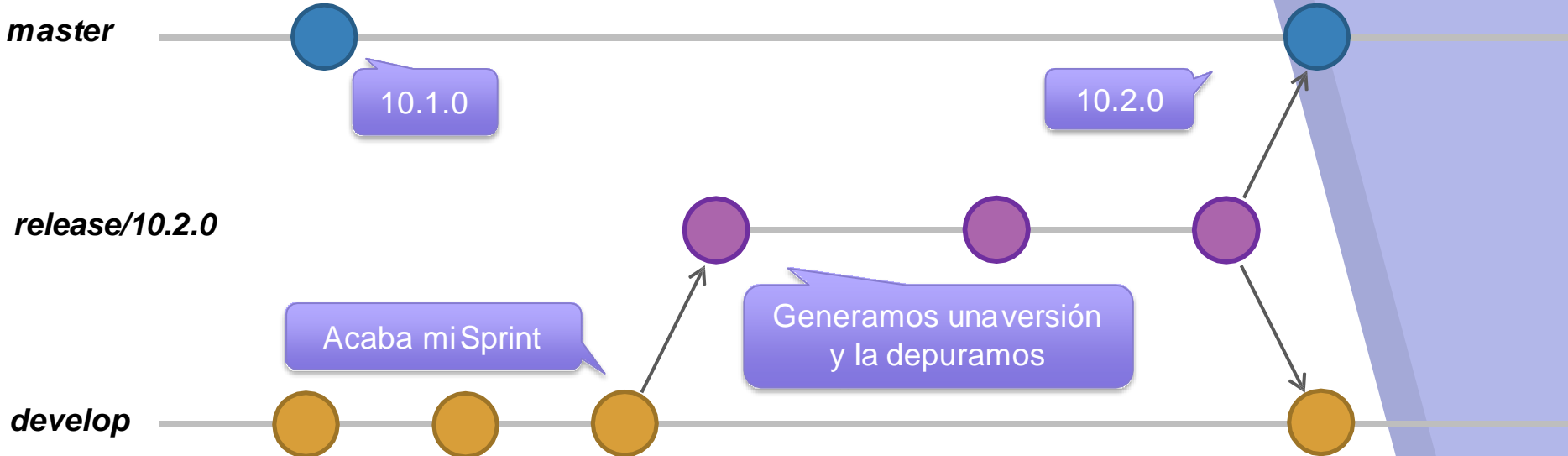
Se genera a partir de develop cuando finaliza el desarrollo de un conjunto de funcionalidades

Su nombre lo define la versión del producto, con release/ al comienzo

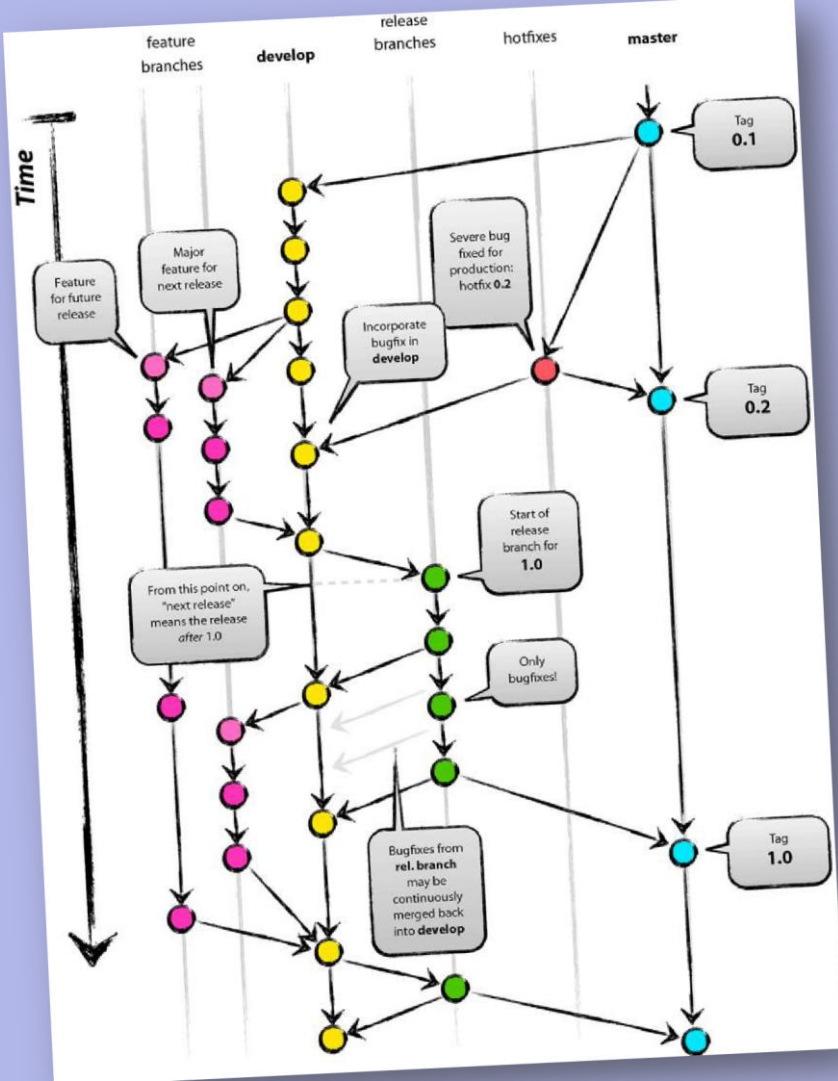
- Nos permite “congelar” y depurar una versión antes de pasar a master

Los pequeños cambios que se requieran pueden hacerse directamente en release

Al final de las fases de prueba o últimos cambios, se mergea con master y develop



UNIÉNDOLO TODO



ADAPTADO A TU ENTORNO



El equipo de QA (Certificación) se sitúa entre el cliente y el equipo de desarrollo, ayudando al cliente a definir los requisitos y los objetivos de las pruebas, y al equipo de desarrollo a verificar correctamente el producto antes de que este llegue al usuario final.