	Proyecto: SADE Migración de SVN a GIT	

Resumen

A continuación se realiza una descripción sobre como migrar SVN a GIT y su impacto en la forma de trabajo

Registro de modificaciones

Versión	Descripción	Autor	Fecha creación	Aprobado por	Fecha aprobación
1.00	Versión inicial	Carlos Bellotti	19/02/2018		
1.10	Agrego escenarios	Carlos Bellotti	22/02/2018		
1.20	Agrego escenarios 3,4 y 5	Carlos Bellotti	27/02/2018		
1.30	Correcciones	Carlos Bellotti	28/02/2018		

Contenido

Introducción

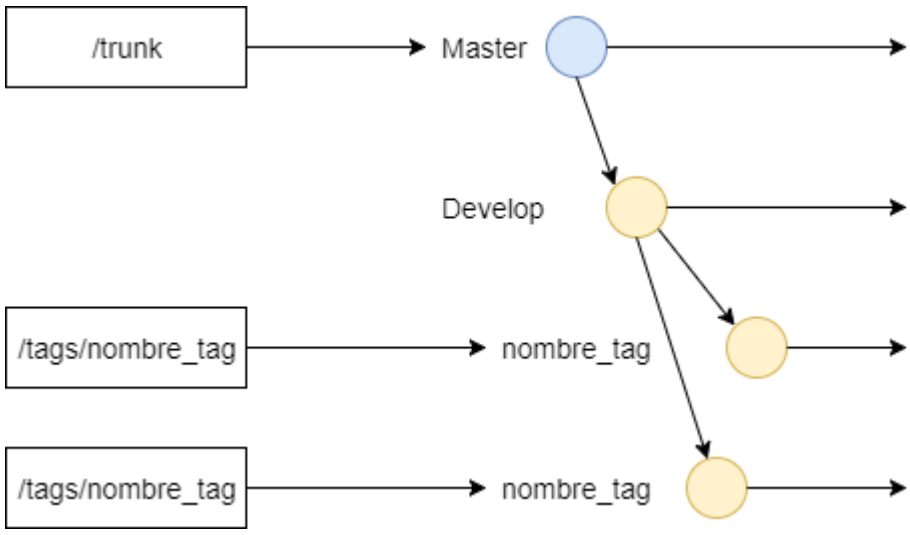
El presente documento tiene como objetivo describir el proceso de migración de SVN a GIT como gestor de repositorios y proponer una metodología de trabajo.

Proceso de Migración

En primera instancia se debe solicitar a todos los equipos de desarrollos que facilite la ruta de SVN de los módulos que mantiene que consideren como master (trunk en SVN) y las ramas que necesitan mantener.

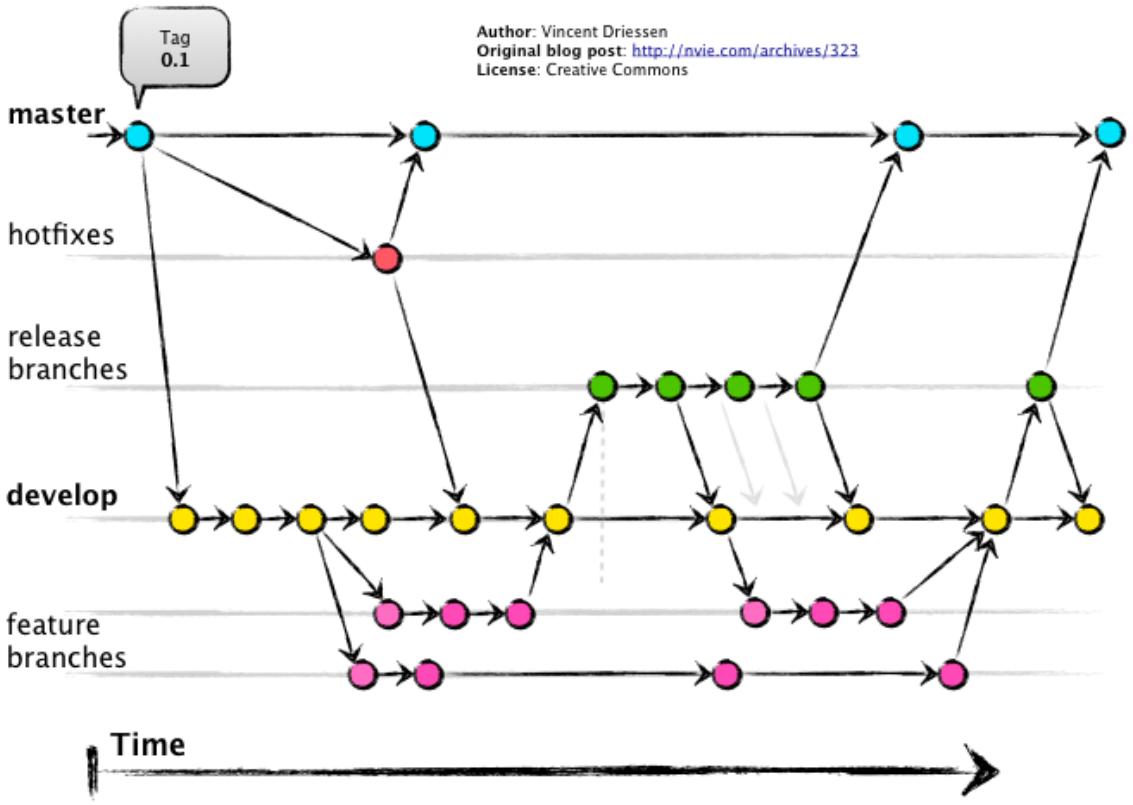
Una vez obtenida esta información se procederá a crear los repositorios en **GitLab** para pasar el código donde se alojara la rama principal de SVN en la rama **master** de GIT.

Luego se crearan las ramas necesarias que mapeen las ramas de SVN que se quieren mantener para que luego se agregue el código adicional correspondiente a cada rama.



Flujo de Trabajo

A continuación se describe el flujo de trabajo propuesto para los desarrollos. El mismo es GitFlow.



En este esquema tenemos dos ramas principales, **master** y **develop**. En un principio ambas son idénticas.

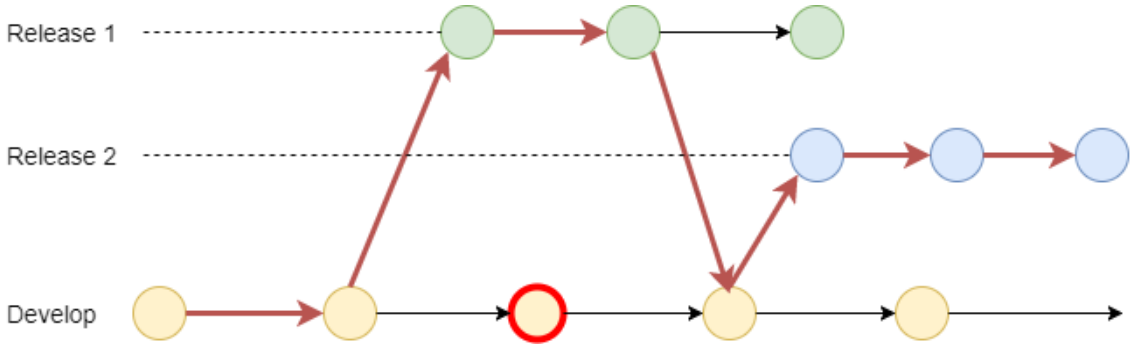
Al inicio de un desarrollo se crea una rama a partir de la rama **develop** donde se realizaran los cambios correspondientes al mismo. Puede haber tantas ramas simultáneas como sea necesario. Una vez se determinan que desarrollos integraran el reléase a enviar al cliente se realiza el merge de las ramas de estos con la rama **develop**. Luego se crea una rama correspondiente al reléase desde **develop** a partir de la cual se generaran los artefactos para entregar al cliente. Cualquier corrección que se necesite realizar se hará sobre esta rama. Una vez el reléase se despliega en producción se realiza el merge de su rama sobre las ramas **master** y **develop** y se realiza un tag indicando la versión.

Todas las ramas, tanto las de desarrollo como las de release una vez realizado el merge deben ser eliminadas.

También durante la certificación del reléase se puede requerir realizar un merge de la rama reléase en curso con **develop** para iniciar un nuevo desarrollo por más que la certificación no haya finalizado aun. En este caso hay que mantener siempre actualizada la rama del desarrollo en curso con respecto a la rama **develop** que puede estar obteniendo el merge final de la rama del reléase como algún merge parcial necesario por alguna corrección sobre este.

Siguiendo estas mecánicas mantenemos la rama **master** copia fiel del código productivo. De esta forma en caso ser necesario realizar una corrección en producción se crea una rama correspondiente a la misma desde **master**, se realiza la corrección, se genera los artefactos necesarios, se realiza el merge de esta rama con **develop** y **master**, se genera el tag y se borra la rama.

En nuestro caso se puede dar que tengamos, para el mismo módulo, un reléase en HML y el reléase siguiente en QA. Cuando se da esta situación se tienen dos ramas de reléase simultaneas y el flujo de creación de la segunda rama debería ser la siguiente (marcado en rojo)



Se puede apreciar como ante la necesidad de realizar un nuevo reléase que contiene el commit marcado en rojo se actualiza la rama **develop** realizando un merge de la rama reléase en esta para luego crear una nueva rama representando el **Release 2**. Téngase en cuenta que de realizarse alguna corrección en el **Release 1** estos deben ser integrados en la rama develop para posteriormente pasarlos a la rama **Release 2.**

Responsabilidades

En el ciclo de vida de desarrollo participaran los siguientes perfiles

- **Líder Técnico:** Es el responsable de mantener la integridad del repositorio realizando las actualizaciones de las ramas activas y eliminando las ramas que ya no son necesarias
- **Desarrollador:** Se encarga de ejecutar las tareas de desarrollo y corrección sobre las ramas creadas por el **Líder Técnico** y es responsable de actualizar su rama de trabajo con las actualizaciones ingresadas en la rama **develop** y verificar que su desarrollo funcione correctamente integrado a las mismas. Esta actualización tendrá una frecuencia a definir con el equipo.

Escenarios

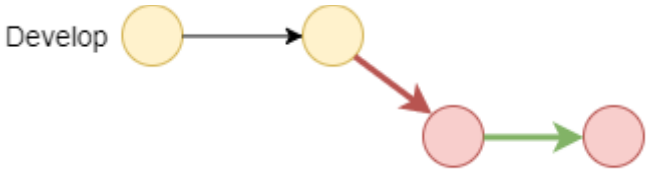
A continuación se presentan una serie de escenarios que describen situaciones comunes

Escenario	Descripción
1	Se presenta la necesidad de realizar un desarrollo
2	Se presenta la necesidad de realizas dos desarrollos en forma paralela para la misma release
3	Se presenta la necesidad de crear una reléase para QA mientras tenemos en curso una reléase en HML
4	Se presenta la necesidad de realizas dos desarrollos en forma paralela que irán en releases distintas
5	Se presenta la necesidad de corregir un bug en producción

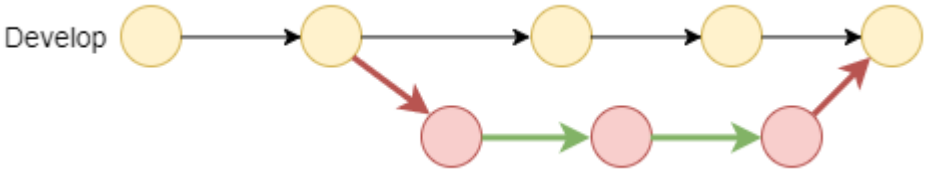
Escenario 1

Se presenta la necedad de realizar un desarrollo.

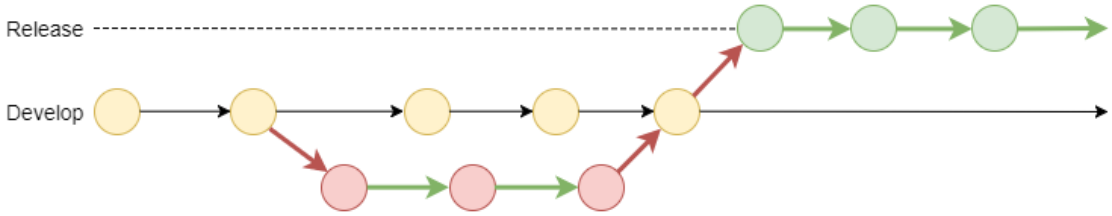
1. El **Líder Técnico** crea una rama a partir de la rama **develop** para el desarrollo a realizar
2. El **Desarrollador** realiza el desarrollo sobre la rama creada



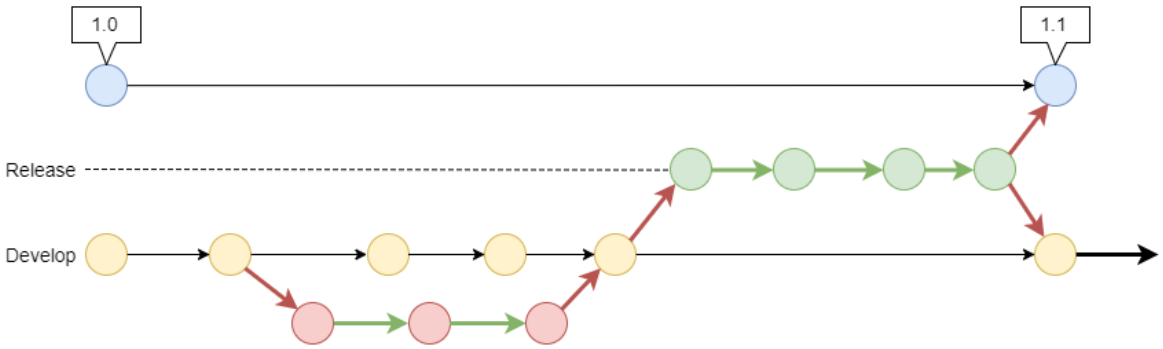
3. Cuando el **Desarrollador** finaliza sus tareas el **Líder Técnico** realiza el merge de la rama en la rama **develop** y la elimina



4. El **Líder Técnico** crea una rama a partir de la rama **develop** para el **reléase**
5. El **Desarrollador** realiza correcciones sobre la rama **develop**



6. Cuando el reléase esta listo se despliega la aplicación y el **Líder Técnico** realiza el merge de la rama de reléase en la rama **develop** y **master**.

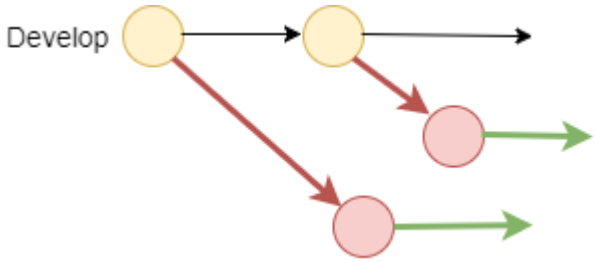


7. Por ultimo genera un TAG en master para la versión y elimina la rama de **release**

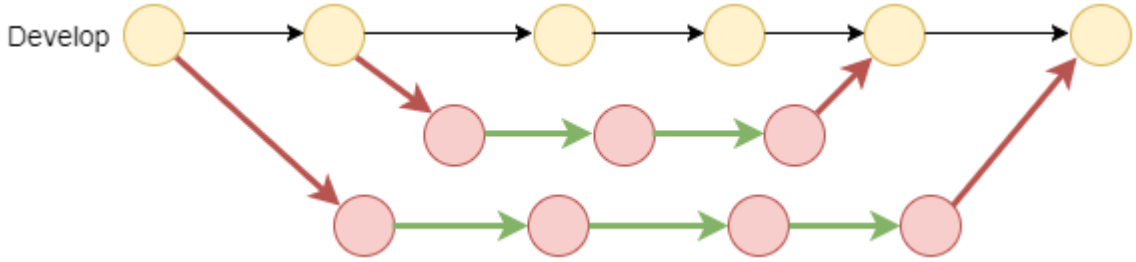
Escenario 2

Se presenta la necesidad de realizar dos desarrollos que irán en la misma reléase

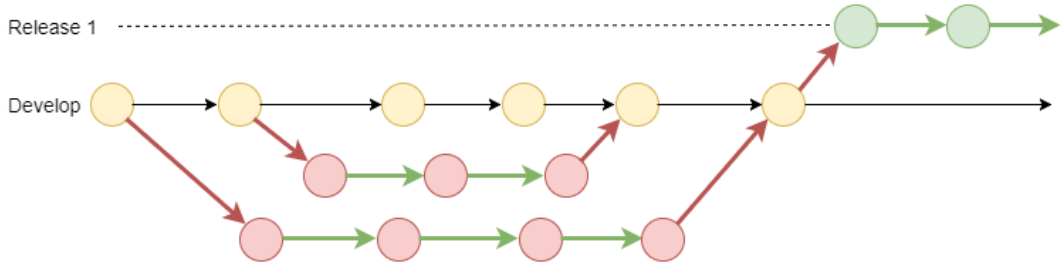
1. El **Líder Técnico** crea las ramas para los desarrollos a partir de la rama **develop**



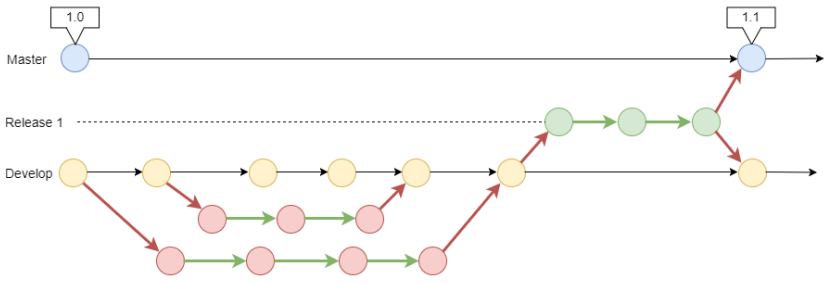
2. Los **Desarrolladores** realizan el desarrollo sobre sus respectivas ramas
3. Una vez los desarrollos estén finalizados el **Líder Técnico** realiza el merge de estos en la rama **develop**



4. El **Líder Técnico** crea la rama reléase a partir de la rama **develop**



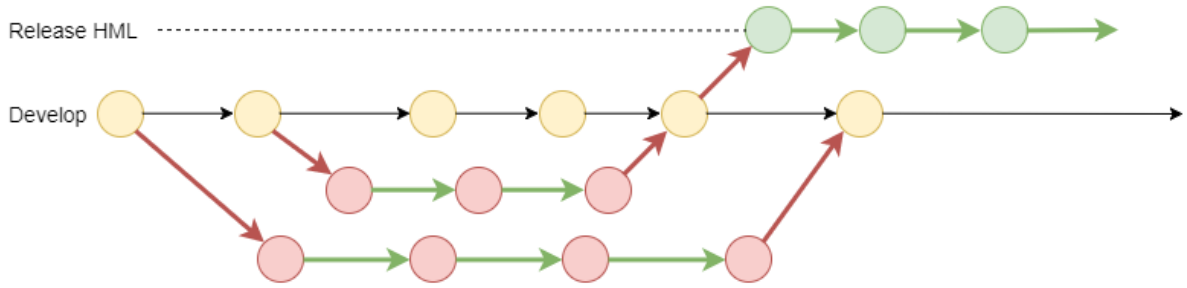
5. Cuando el reléase se encuentra productivo el **Líder Técnico** realiza el merge de la rama de reléase en las ramas **master** y **develop**.



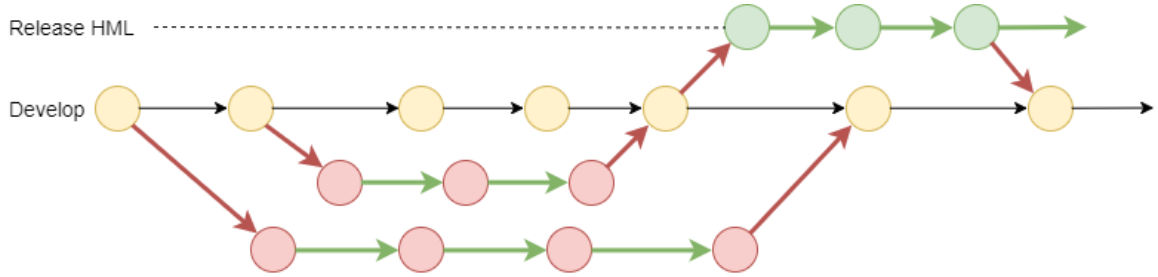
Escenario 3

Se presenta la necesidad de enviar un reléase a QA teniendo un reléase en HML.

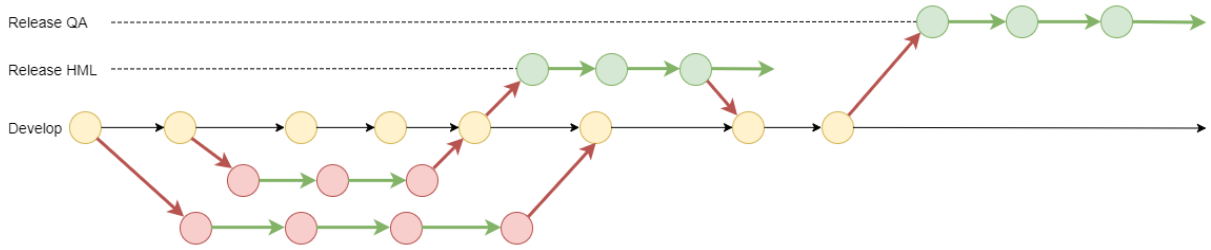
1. El **Líder Técnico** realiza un merge de la rama de desarrollo a integrar en nuevo reléase con la rama **develop**



2. El **Líder Técnico** realiza un merge de la rama reléase (Rama HML) en curso en la rama **develop**

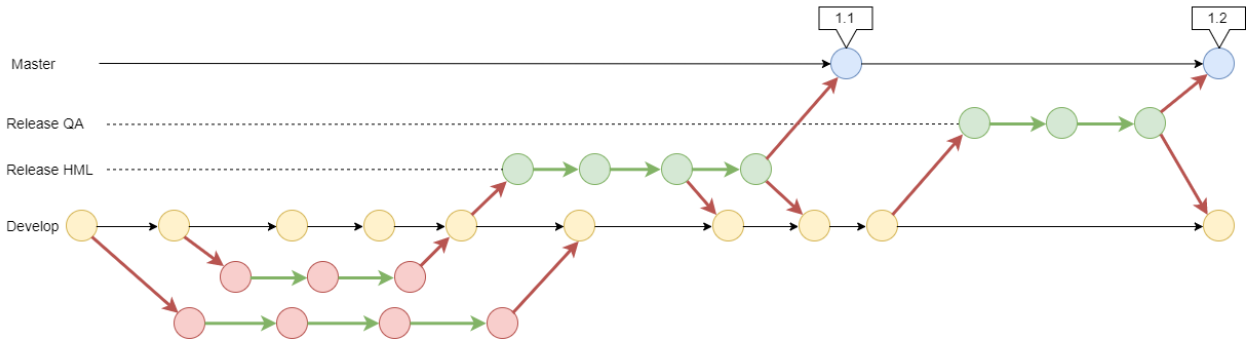


3. El **Líder Técnico** genera una nueva rama para el nuevo reléase (Rama QA)



4. Se generan los artefactos necesarios a partir de esta nueva rama
5. Si se llega a producir una corrección en la rama HML
    a. El **Líder Técnico** debe realizar el merge de esta rama en la rama **develop** para actualizarla
    b. Luego debe realizar merge de la rama **develop** en la rama QA
6. De ser necesario el **Desarrollador** realiza correcciones sobre la rama QA

7. Si la rama de QA llega a producción en algún momento se realizan los pasos 6 y 7 des Escenario 1.

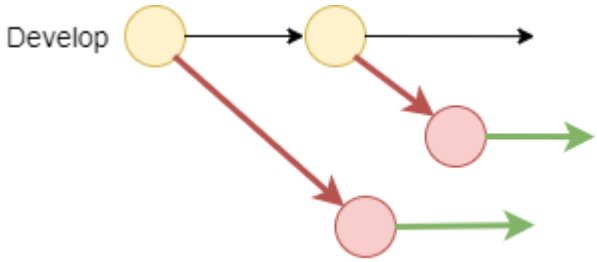


8. Si la rama de QA no llega individualmente a producción sino que forma parte de un reléase posterior el **Líder Técnico** debe realizar el merge de esta rama en la rama **develop** para iniciar un nuevo reléase que sume las características y correcciones de la rama QA mas los nuevos desarrollos a enviar con el reléase.

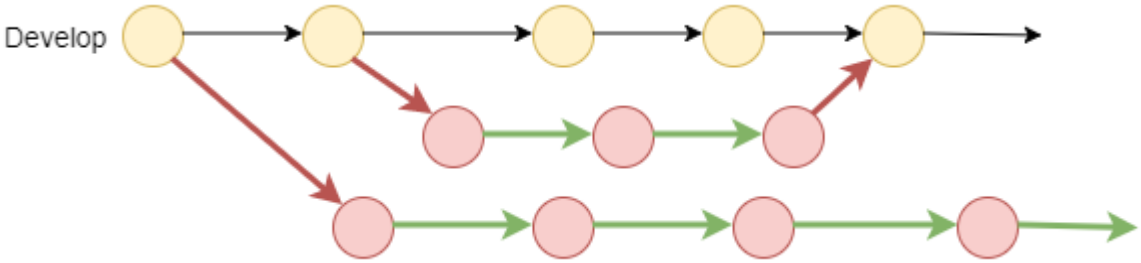
Escenario 4

Se presenta la necesidad realizar dos desarrollos en paralelo que irán en releases distintas.

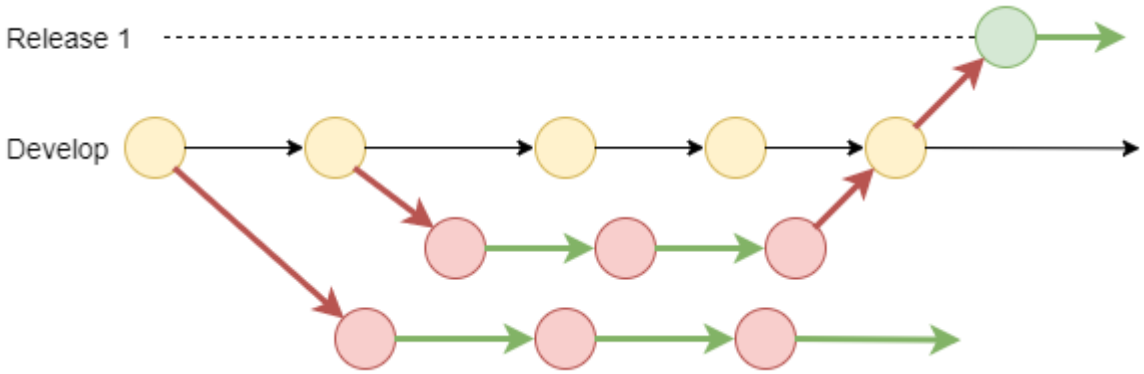
1. El **Líder Técnico** crea las ramas para ambos desarrollos a partir de la rama **develop**



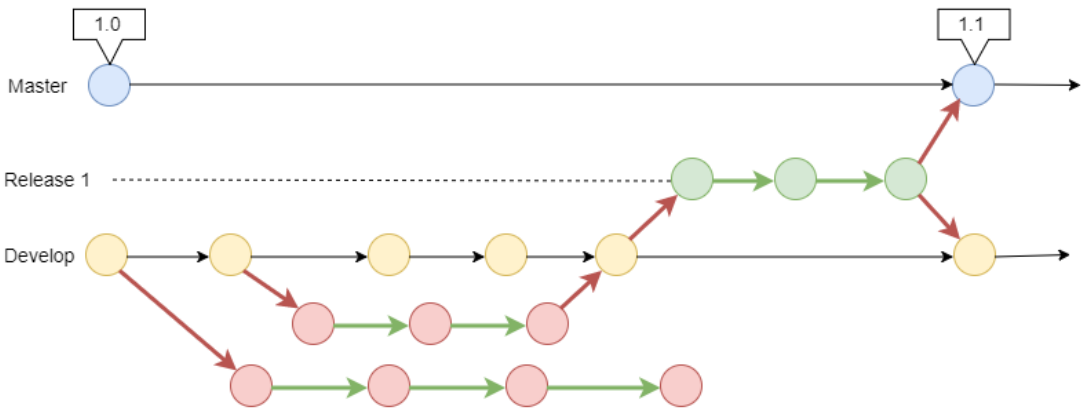
2. Los **Desarrolladores** realizan su trabajo en las ramas que representan los desarrollos
3. El **Líder Técnico** realiza el merge de la rama correspondiente al reléase 1 en la rama **develop**



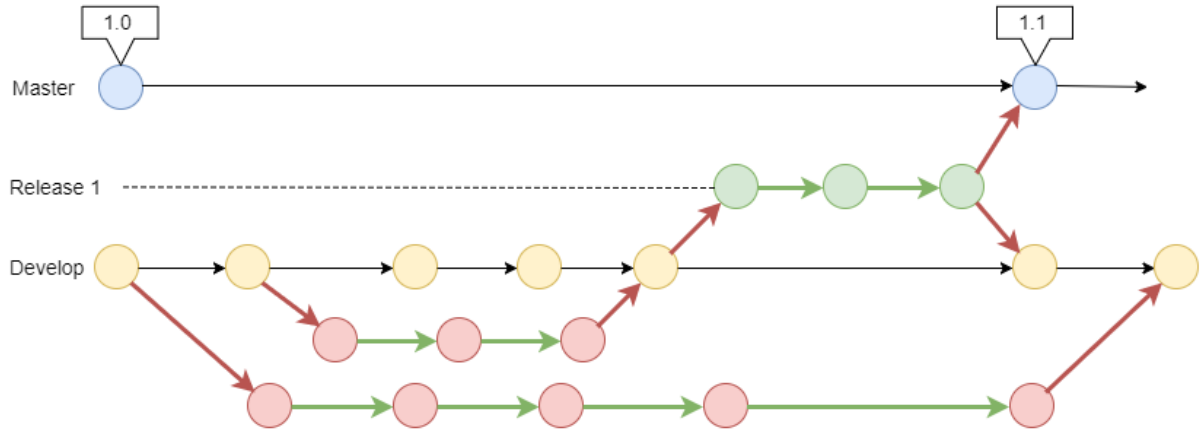
4. El **Líder Técnico** crea la rama reléase 1 a partir de la rama **develop**



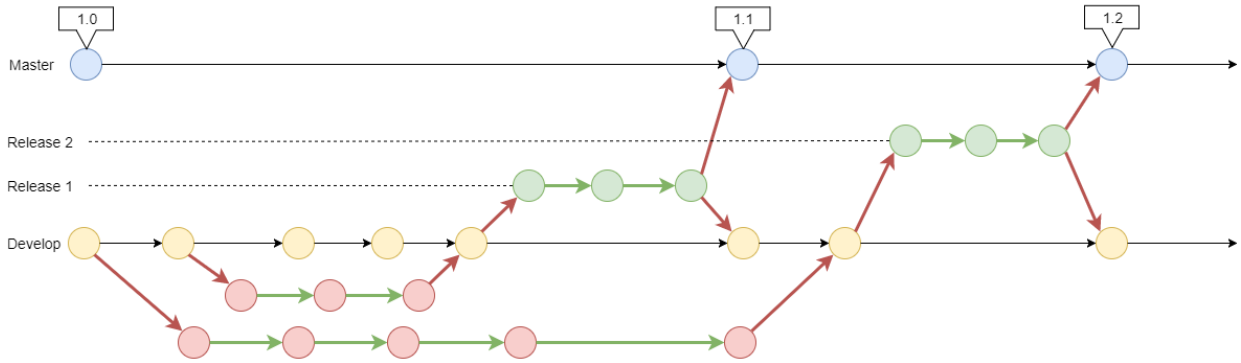
5. Una vez esté productivo el reléase 1 el **Líder Técnico** realiza el merge de la rama reléase 1 en la rama **develop** y en la rama **master** y borra la rama de reléase 1.



6. Cuando el desarrollo correspondiente al reléase 2 está listo el **Líder Técnico** realiza el merge de la rama del desarrollo en la rama **develop**



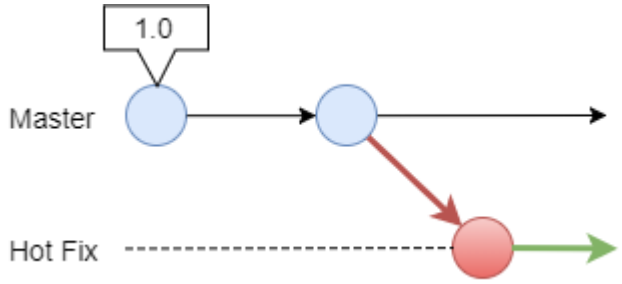
7. El **Líder Técnico** crea la rama reléase 2 a partir de la rama **develop**
8. Una vez el reléase 2 esté productivo el **Lider Tecnico** realiza el merge de su rama en las ramas **develop** y **master**



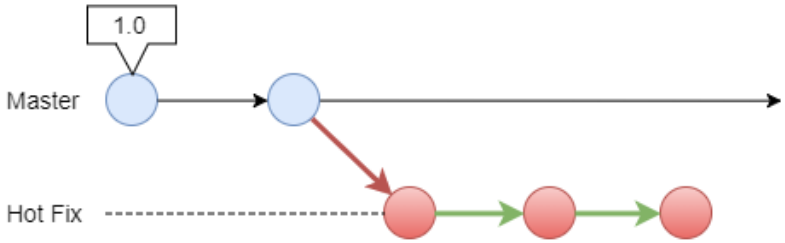
Escenario 5

Se presenta la necesidad de corregir un bug en producción.

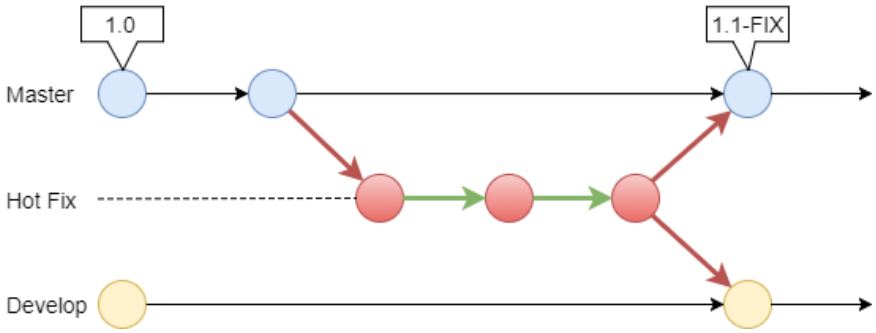
1. El **Líder Técnico** crea una rama para realizar la corrección a partir de la rama **master**



2. El **Desarrollador** realiza las correcciones sobre la rama creada



3. Una vez este la corrección en producción el **Líder Técnico** realiza el merge de la rama de esta en las ramas **master** y **develop**



4. Si hay ramas de reléase en curso el **Líder Técnico** debe realizar el merge de la rama **develop** con las ramas de reléase para que las mismas contengan la corrección.

Herramientas

Git se puede operar utilizando la línea de comando o en su defecto TortoiseGit, muy similar al Tortoise SVN.

Git: https://git-scm.com/

TortoiseGit: https://tortoisegit.org/download/

Guia Rapida: http://www.edy.es/dev/docs/git-guia-rapida/