



NTT Data

SISTEMAS DE CONTROL DE VERSIONES

FORMACIÓN SEVILLA - MÁLAGA

Resumen

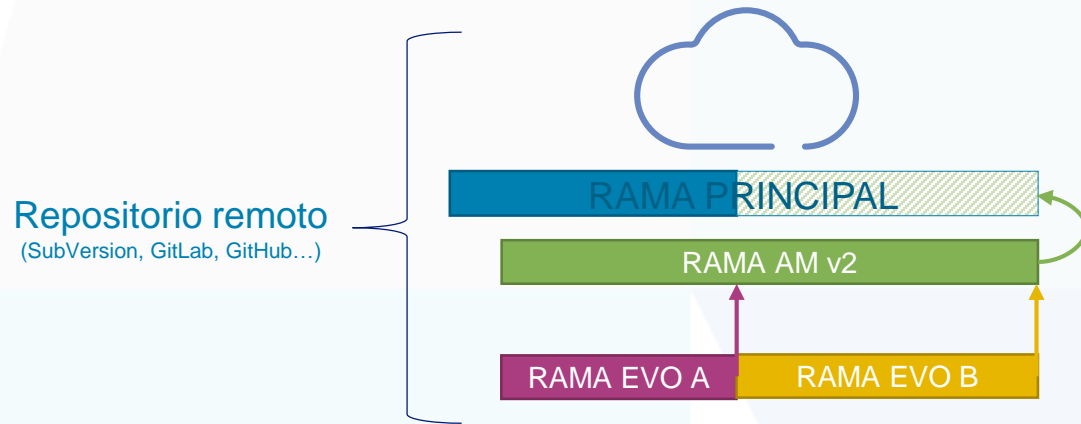
**FUTURE
AT HEART**

SISTEMA DE CONTROL DE VERSIONES

- Gestión de ficheros a lo largo del tiempo.
- **Versionado** del código fuente.
- Permite el **trabajo remoto y colaborativo**.
- Dos tipos:
 - **Distribuidos** → GIT como referencia.
 - **Centralizados** → SubVersion (SVN) como referencia.



CONCEPTO RAMIFICACIÓN



Ejemplo de ramificación:

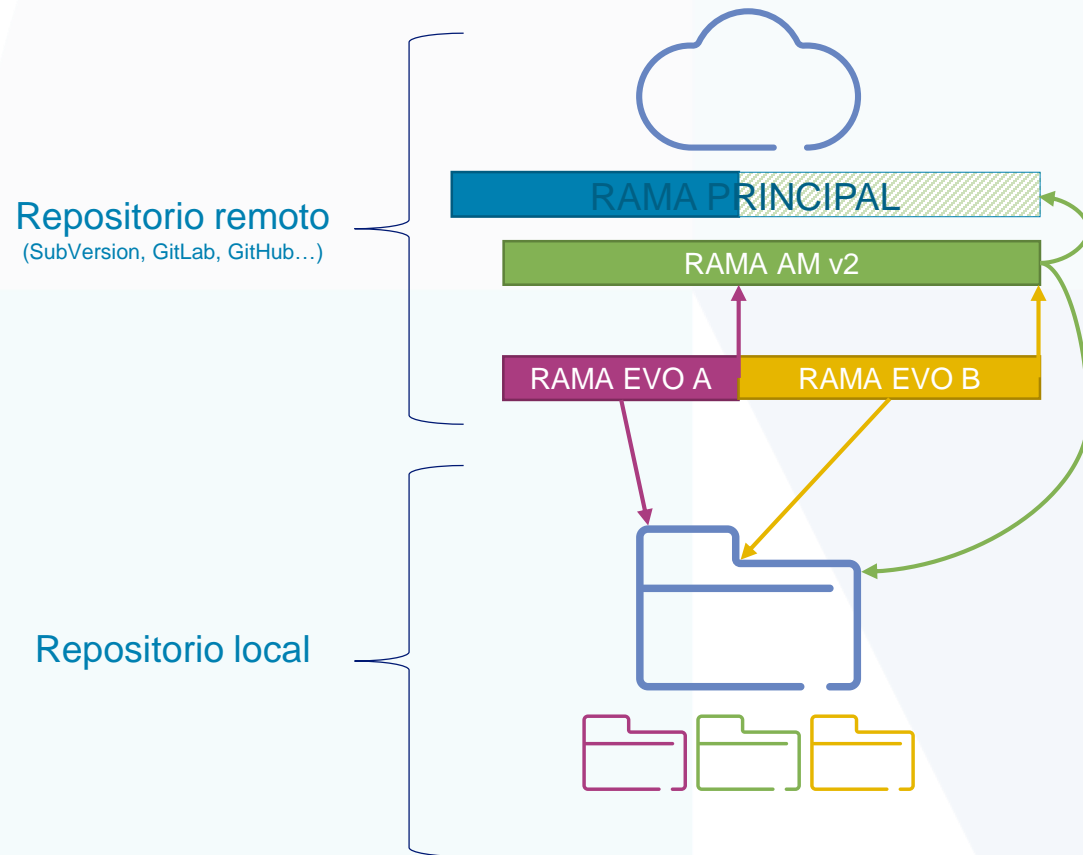
- Rama principal (producto final)
- Rama de correctivos (AM)
- Rama de evolutivos (A y B)

La rama principal recibe periódicamente integraciones de ramas AM que aumentan la principal a lo largo del tiempo. Las ramas de evolutivos tienen ciclos más cortos y van integrándose en la rama de correctivos. Garantizada la viabilidad de la rama AM se integrará en la principal (main, master, trunk...)

Otro posible ejemplo sería en el que las ramas evolutivas estuviesen al mismo nivel que la de correctivos (AM) de forma en que las integraciones con la rama principal fuesen directas.

La integración o merge es el proceso de volcar una rama en otra (merge también es utilizado a nivel de clases en una misma rama)

CONCEPTO RAMIFICACIÓN

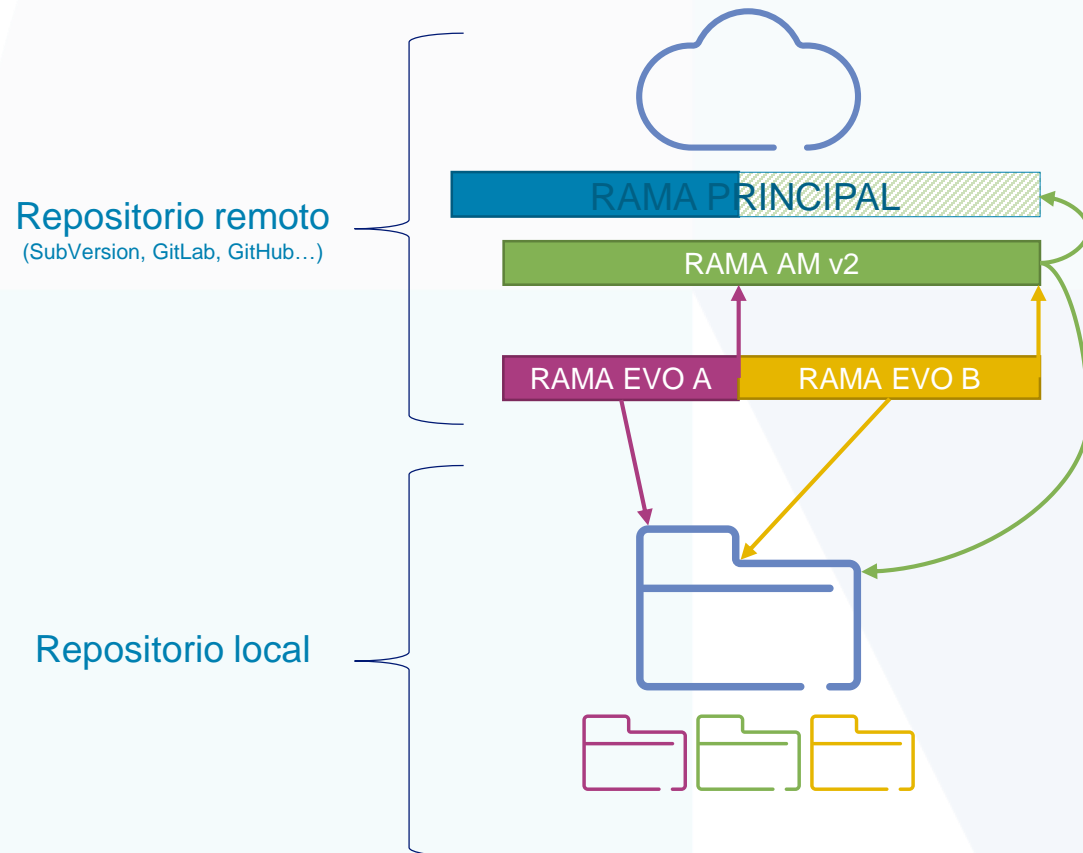


En el repositorio local es posible descargar tantas ramas como se quieran, siendo cada una de éstas el propio repositorio representado como un directorio de nuestro sistema operativo.

Sólo con GIT es posible ramificar también en los repositorios locales.

Toda la información del repositorio local de GIT, ramas, versionado... se almacena en el directorio `.git`.

BUENAS PRÁCTICAS



- Mantener el repositorio local actualizado con relativa frecuencia (1 o 2 dos veces al día) de forma en la que se reduzca el volumen de conflictos ocurridos, así como la gravedad o complejidad de resolución del mismo, en el proceso de integración o merge.
- Ante un conflicto, tener en cuenta que siempre habrá que tomar una decisión entre tres posibles, siendo la tercera la opción más coherente si se sigue una buena línea de trabajo (comunicación y metodología técnica):
 - a) Elegir que prevalezca el código remoto.
 - b) Elegir que prevalezca el código local.
 - c) Elegir que prevalezca la mezcla del código local y remoto.
- Tener en cuenta que el workspace del IDE no tiene porque ser el repositorio local de código. De hecho, ésta opción no es aconsejable en proyectos complejos. Lo ideal es tener un directorio propio para ubicar el código fuente que ejerza de repositorio local ajeno al workspace.
- Mantener comunicación fluida con el equipo de desarrolladores/as.

ETAPAS CLAVES (SCV)

SVN:

- REPOSITORIO LOCAL → ADD → COMMIT (VERSIONA) → REPOSITORIO REMOTO.

GIT:

- REPOSITORIO LOCAL → ADD → STAGING AREA → COMMIT (VERSIONA) → PUSH → REPOSITORIO REMOTO.

NTT DATA

GRACIAS

**FUTURE
AT HEART**

