

# Git

Uso de Git

# SCM vs VCS

VCS: Version Control System/Software

Es lo mismo, yasta.

# SCM

## FUNCIONES

# Almacenamiento Centralizado

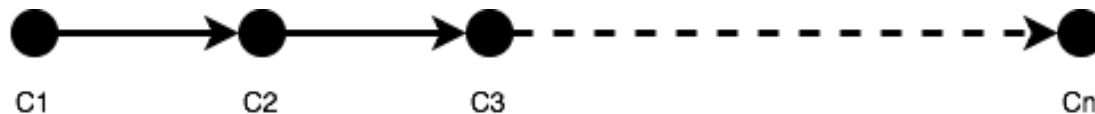
Nos aporta:

- Origen de Verdad
- Copias de Seguridad de Desarrolladores
- Copias de Seguridad del Proyecto

¿Podemos solucionarlo con un almacenamiento compartido en red?  
FTP, NAS, OneDrive, S3, ...

# Registro del Cambio = Versiones

Cada modificación en el contenido del proyecto queda almacenada, de forma que es posible recuperar el estado (snapshot) del proyecto a lo largo de toda la historia de su evolución.



A cada cambio lo vamos a llamar “commit”

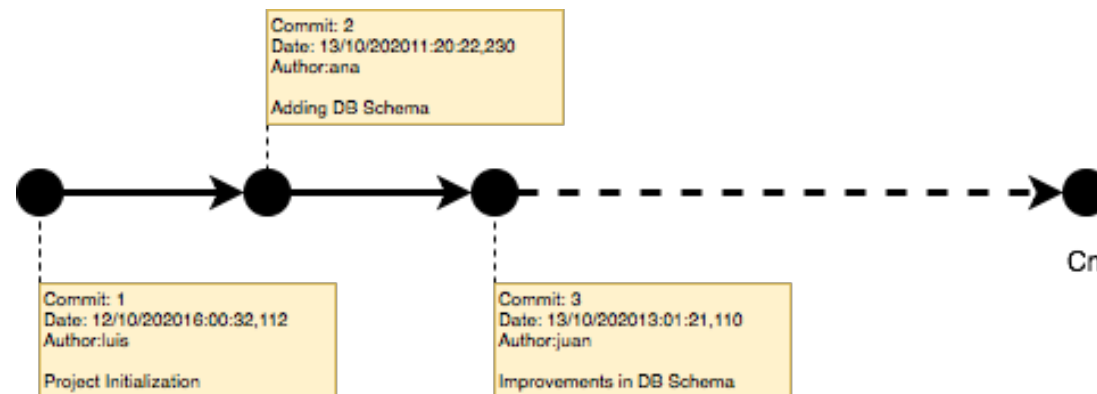
El proyecto tiene una raíz (un directorio) del que cuelgan, jerárquicamente, todos los artefactos

Cada commit almacena una copia completa del proyecto=directorio.

# Auditoría del Cambio - Trazabilidad

En cada commit se registra meta-información sobre:

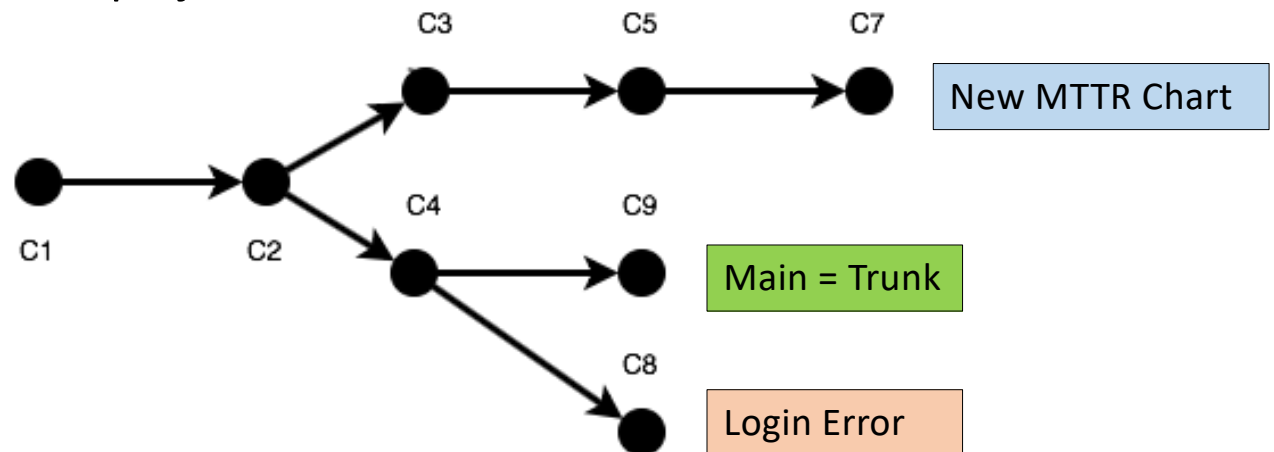
- Cuando se hizo: Fecha y hora
- Quién lo hizo: Usuario del SCM que registró el cambio
- (Opcional) Por qué se hizo: Descripción libre del autor del cambio



# Ramas - Branches

Registra desarrollo paralelos del Proyecto

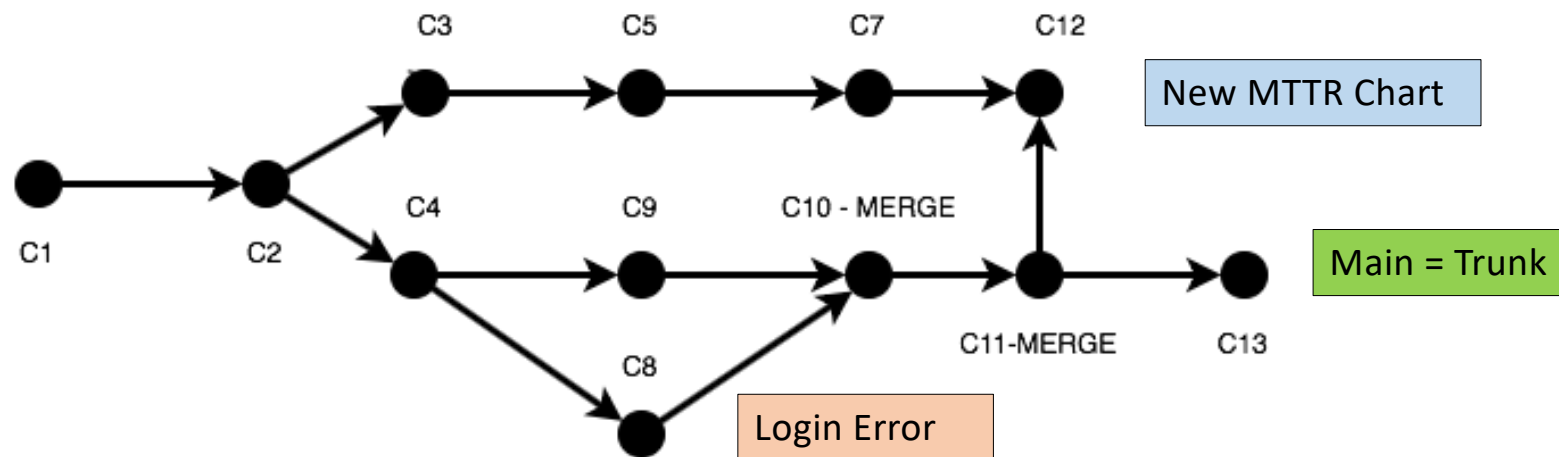
- Bug Fixing en versiones liberadas
- Desarrollos Independientes:
  - Nuevas Funcionalidades Complejas
  - Pruebas de Concepto



# Mezcla - Merge

Integra los cambios

1. La solución del bug se integra en Trunk
2. El desarrollador de MTTR Chart incorpora los cambios de Trunk
3. El desarrollo de MTTR Chart se descarta





# Branches

Permite paralelizar al aislar los trabajos paralelos de los sub-equipos.

Mejora la productividad

Introduce duplicidad, inconsistencia que habrá que solucionar.

Acumulamos Deuda

## ¿Compensa?

# Estrategias de Branches

Feature Branches

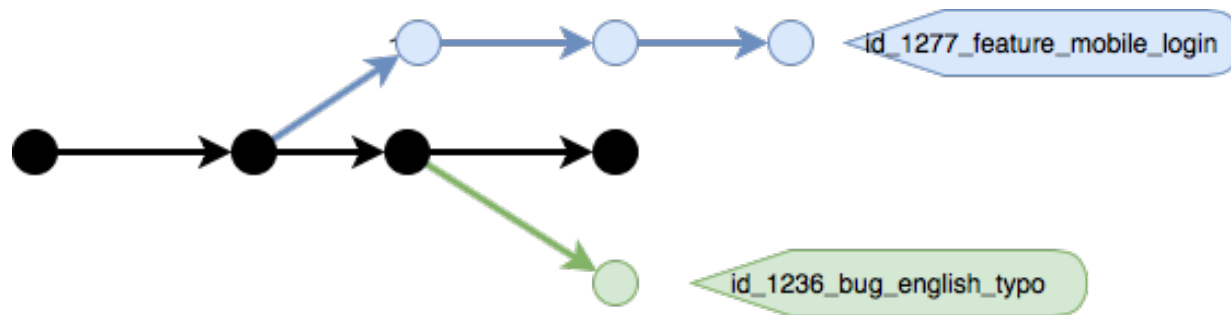
Git Workflow

Trunk Based Development

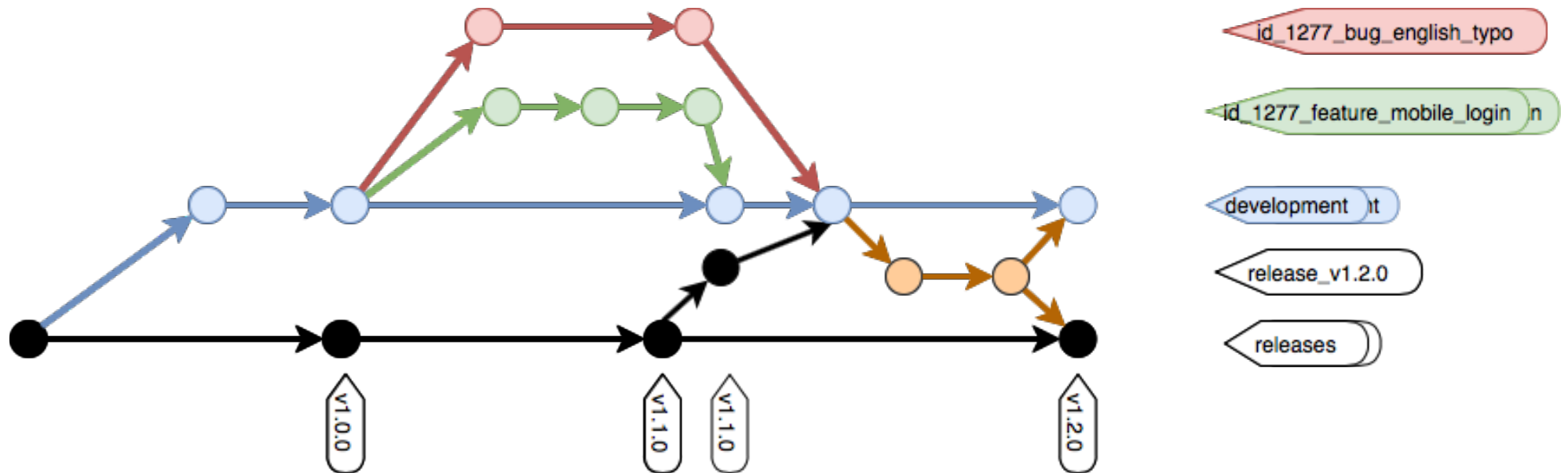
# Feature Branches

Creamos una Rama para desarrollar cada cambio en la aplicación: Story, Task, Activity.

- Etiquetamos la Rama con ítem que se desarrolla: <id>\_<name>



# Git Workflow



# Archivos de Texto vs Binarios

Los SCM registran las diferencias en:

- Sistema de Ficheros

- Archivos de Texto

El versionado de Binarios se realiza mediante copia.

- Existen SCMs específicos para gestionar formatos binarios conocidos: vídeo, imágenes, etc.

# Monorepos

Llevamos al límite los principios de transparencia y sharing.

## Inconvenientes:

- Pesado
- Derechos de acceso

## Soluciones:

- Plugins para GIT
- SVN



# SCM

Soluciones

# Repositorio Centralizado

Cada commit es compartido: máxima integración.

Necesitas conexión para registrar cambios o consultar el histórico

El conflicto se evita: bloqueo de archivos

Administración centralizada: celos

CVS (no tiene continuidad)

Subversion





# Repositorio Distribuido

Cada desarrollador tiene una copia INDEPENDIENTE del repositorio  
(Intrínsecamente descentralizado. )

Se comparte en dos fases: Registro Commit local y subo a Repositorio acordado como Compartido.

Permite registrar cambios locales: depurarlos, sin conexión.

Git

Mercurial

Bazaar



# Por qué GIT

Es el estándar de facto.

- Disponible en Plataformas
- Probado y fiable
- Se integra con múltiples soluciones

Prestaciones. Es ligero y optimiza el almacenamiento diferencial.

Consistente. El registro de cambios es a nivel de proyecto (simple) y la integridad se garantiza con técnicas de cifrado.

No impone condiciones sobre la estrategia de desarrollo

