

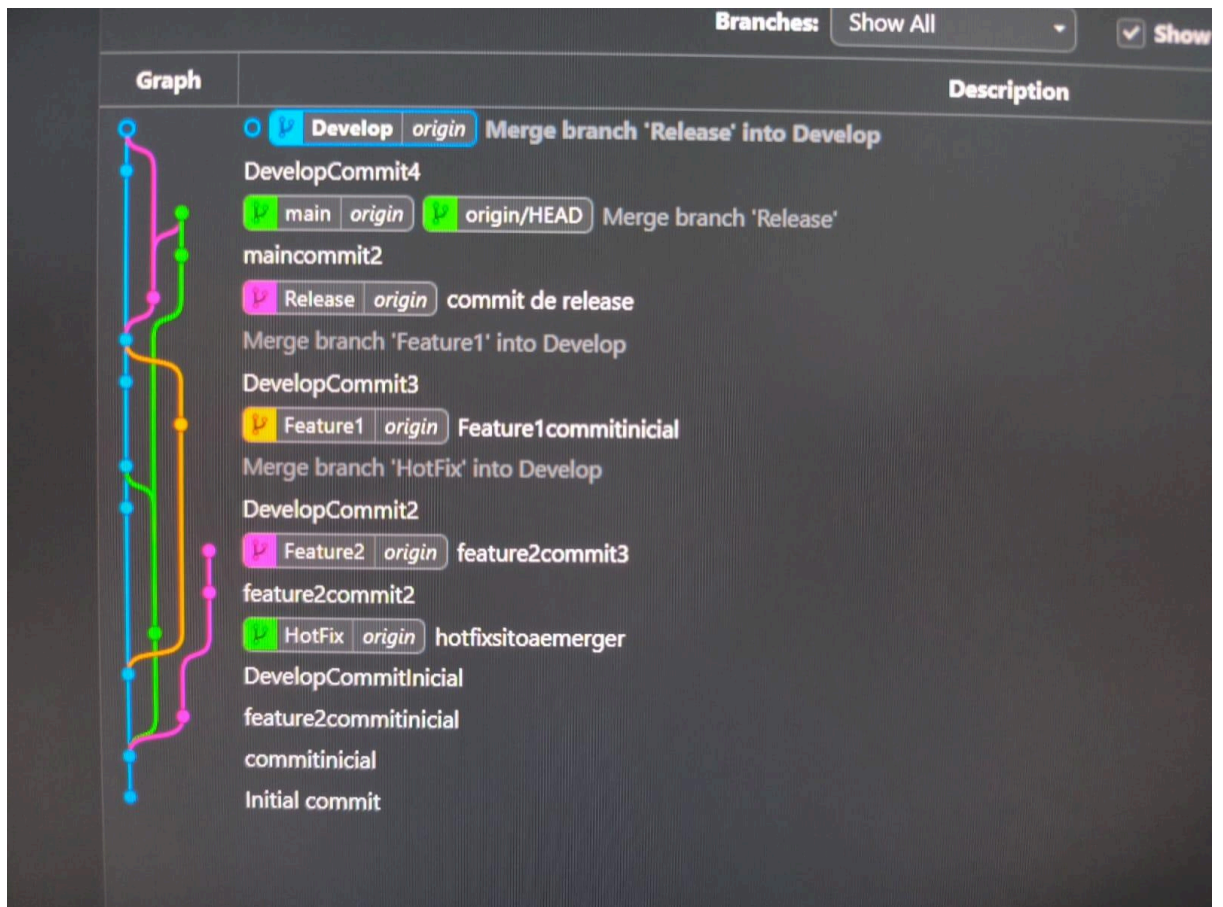
# **Sistemas de Repositorios Tema 18**

Rafael Ángel Alvariño Jiménez



# **Documentación utilizada para crear las ramas en el repositorio mencionada en el readme**







# Sistemas Centralizados VS Distribuidos

Tanto los sistemas de control de versiones centralizados como los distribuidos poseen diferentes ventajas y desventajas. La elección del más adecuado, por tanto depende de las necesidades y preferencias del equipo de desarrollo.

Subversion y Git son los dos ejemplos más populares de sistema de control de versiones funcionando el primero mediante un sistema centralizado y otro mediante uno distribuido.

Normalmente usaremos uno de esos dos dependiendo del sistema que queramos.

Ventajas de los sistemas centralizados (Subversion):

- Simplicidad: La estructura centralizada ofrece un modelo más sencillo de comprender y usar, especialmente para equipos nuevos o con poca experiencia en sistemas de control de versiones.
- Fácil Administración: Un único repositorio centralizado facilita la administración y el control de las versiones del código, especialmente en entornos con políticas de acceso estrictas o flujos de trabajo jerárquicos.
- Facilidad de auditoría: El historial centralizado de cambios facilita la auditoría y el seguimiento de las modificaciones realizadas en el código.

Las Desventajas de los sistemas centralizados son:



Dependencia del servidor central: El acceso y la modificación del código dependen de la disponibilidad del servidor central, lo que puede generar cuellos de botella o problemas de acceso en caso de caídas o interrupciones.

Falta de acceso sin conexión: La ausencia de copias locales completas del repositorio limita el trabajo sin conexión a internet.

Escalabilidad limitada: A medida que crece el tamaño del repositorio y del equipo, la gestión del servidor central puede volverse más compleja y costosa.

Por otro lado, tenemos los sistemas distribuidos como git cuyas ventajas son:

Trabajo sin conexión: Cada desarrollador posee una copia completa del repositorio, lo que permite trabajar sin conexión a internet y facilita la colaboración en ubicaciones remotas o con conectividad intermitente.

Flujo de trabajo ramificado: Git facilita la creación y gestión de ramas independientes, lo que promueve la experimentación y el desarrollo simultáneo de diferentes características o correcciones.

Buena capacidad de escalabilidad: La arquitectura distribuida, al no depender de un único servidor centralizado, es más resistente a fallos y permite una mejor escalabilidad a medida que crece el proyecto. También se puede trabajar de forma más ágil y el acceso al servidor y a los datos, que siempre almacenan una copia es más rápido.

Historial descentralizado: Cada copia local del repositorio posee un historial completo de cambios, lo que facilita la resolución de conflictos y la recuperación de versiones anteriores.

Por otro lado sus desventajas son:



Curva de aprendizaje más pronunciada: La complejidad de la estructura distribuida y la ausencia de un servidor centralizado puede requerir mayor práctica para los equipos nuevos.

Mayor responsabilidad individual: Cada desarrollador es responsable de mantener la integridad y coherencia de su copia local del repositorio, lo que exige mayor disciplina y coordinación.

Gestión de fusiones más compleja: La combinación de ramas y la resolución de conflictos de fusión pueden ser más complejas en comparación con un sistema centralizado.

Conclusión:

La elección entre un sistema centralizado como Subversion o uno distribuido como Git depende de varios factores, como el tamaño del proyecto, la experiencia del equipo, la necesidad de trabajo sin conexión, la importancia del control centralizado y la complejidad del flujo de trabajo.

Subversion es una buena opción para equipos pequeños o con poca experiencia en VCS que valoran la simplicidad, el control centralizado y la facilidad de auditoría.

Git, por otro lado, se recomienda para equipos más grandes o con flujos de trabajo complejos que requieren trabajo sin conexión, flexibilidad en la ramificación, robustez y escalabilidad.

En la actualidad, Git es mucho más usado debido a que justamente en las grandes empresas se necesita esa capacidad extra que otorga GIT respecto de subversion y, dado que hablamos de una empresa, incluso si es pequeña, se espera que los trabajadores que participen en un proyecto estén suficientemente cualificados (o adquieran esa cualificación) como para usar Git de manera que una buena gestión del proyecto evite que las desventajas de git sean relevantes.