

4.B. Control de versiones

3. Estructura de las herramientas de control de versiones.

Una **herramienta de control de versiones** es un **sistema de mantenimiento de código fuente** (o de grupos de archivos en general) de gran **utilidad** para grupos de **desarrolladores** que trabajan **cooperativamente**.

Los clientes pueden actuar simultáneamente sobre los mismos ficheros, **comparar diferentes versiones de ficheros**, solicitar una **historia completa de los cambios**, o sacar una "foto" histórica del proyecto tal como se **encontraba en una fecha o versión determinada**.

Los sistemas de control de versiones suelen estar formados por un **conjunto de componentes**:

- **Repositorio**: es el lugar de almacenamiento o base de datos de los proyectos.
- **Rama**: revisiones paralelas del contenido del proyecto para efectuar cambios sin tocar la evolución principal (que normalmente se refiere como rama principal o trunk).
- **Versión**: fotografía del estado de una de las revisiones de alguna rama del repositorio en un determinado momento en el tiempo.
- **Revisión**: evolución en el tiempo de las ramas del proyecto.
- **Etiqueta**: información textual que se añade a un conjunto de archivos o a un módulo completo para indicar alguna información importante. En ocasiones identifican versiones del producto ya cerradas.

Algunos de los **servicios** que típicamente proporcionan son:

- **Creación de repositorios**. Creación del esqueleto de un repositorio sin información inicial del proyecto.
- **Clonación de repositorios**. La clonación crea un nuevo repositorio y vuelca la información de algún otro repositorio ya existente. Crea una réplica.
- **Descarga de la información del repositorio principal al local**. Sincroniza la copia local con la información disponible en el repositorio principal.
- **Carga de información al repositorio principal desde el local**. Actualiza los cambios realizados en la copia local en el repositorio principal.
- **Gestión de conflictos**. En ocasiones, los cambios que se desean consolidar en el repositorio principal entran en conflicto con otros cambios que hayan sido subidos por algún otro desarrollador. Cuando se da esta situación, las herramientas de control de versiones tratan de combinar automáticamente todos los cambios. Si no es posible sin pérdida de información, muestra al programador los conflictos acontecidos para que sea éste el que tome la decisión de como combinarlos.
- **Gestión de ramas**. Creación, eliminación, integración de diferencias entre ramas, selección de la rama de trabajo.
- **Información sobre registro de actualizaciones**.
- **Comparativa de versiones**. Genera información sobre las diferencias entre versiones del proyecto.