

## 4.C. GIT

### 1. Introducción.

En un proyecto de trabajo colaborativo, cada miembro del equipo realiza las tareas que le han sido asignadas, pero al terminar cada uno de ellos, habrá que preguntarse:

- ¿Cómo se integran todas las partes?.
- Si hay errores o cambios, ¿cómo se actualizan los nuevos cambios?.
- Si hay una nueva versión, ¿cómo se gestionan los conjuntos de módulos compatibles con cierta versión?, ¿cómo se recuperan versiones anteriores?, ¿cómo se fusionan versiones? ...

En el libro [Control de versiones](#) se hace referencia a diferentes herramientas de control de versiones, aquí se tratará de mostrar como una de ellas ([GIT](#)) da respuesta a las preguntas anteriores.

Algunas de las características de **GIT** son:

- **Gratis**, de código abierto.
- **Muy popular**, disponible en múltiples plataformas.
- **Distribuido**. Las diferentes réplicas de los repositorios tienen una relación de igual a igual. Esta es una consideración técnica, en la práctica una de ellas suele adquirir el rol de repositorio principal.
- **Basado en changesets**. Si varios componentes del proyecto han cambiado respecto a la última versión, todos ellos son volcados al repositorio de una vez, esta entrega se considera atómica.
- **No bloqueante**. Es posible que varios repositorios locales estén trabajando de forma simultánea sobre los mismos componentes, posteriormente habrá que integrar las modificaciones efectuadas en todos ellos para obtener un versionado común.