4.B. Control de versiones

1. Introducción.

Cuando estamos desarrollando software, el código fuente está cambiando continuamente, bien por el propio desarrollo o por el mantenimiento a que se ve sometido. Además los proyectos en ocasiones se desarrollan por fases o se hacen diferentes entregadas al cliente. Todos estos factores hacen necesario un **sistema de control de versiones**.

Las ventajas de utilizar un sistema de control de versiones son múltiples. Un sistema de control de versiones bien diseñado **facilita** al equipo de **desarrollo** su labor, permitiendo que varios programadores trabajen en el mismo proyecto (incluso sobre los mismo archivos) de forma simultánea, permitiendo gestionar los conflictos que se puedan producir por actualizaciones simultáneas sobre el mismo código. Las herramientas de control de versiones proveen de un sitio central donde almacenar el código fuente de la aplicación, así como el historial de cambios realizados a lo largo de la vida del proyecto.

También permite a los desarrolladores volver a versiones estables previas del código fuente si es necesario.

Una **versión**, desde el punto de vista de la evolución, se define como la forma particular de un objeto en un instante o contexto dado. Se denomina **revisión**, cuando se refiere a la evolución en el tiempo.

Pueden coexistir varias versiones alternativas en un instante dado y hay que disponer de un método, para designar las diferentes versiones de manera organizada.

Existen distintas alternativas de control de versiones de código abierto, **GIT, CVS, Subversion, Mercurial...** En ocasiones podrán ser motivo de gestión proyectos software en desarrollo o simplemente conjuntos de archivos de algún otro uso.

Un repositorio interesante para la gestión de proyectos es **Bitbucket**, está basado en Git y dispone de una versión gratuita para equipos de 5 personas. Ideal para gestionar el proyecto fin de ciclo.