

Unidad didáctica 3:

Control de versiones. Conceptos teóricos.



Módulo: Entornos de Desarrollo
Profesora: Estefania Monerri Mínguez



Introducción

Para un trabajo que no comienza y termina dentro de un período corto de tiempo, o para trabajos que deben ser desarrollados por un equipo de personas, es recomendable tener un control de las tareas que se han realizado, quien las ha hecho...

Se habla de gestión de proyectos cuando se debe desarrollar un trabajo como el que se acaba de definir y, además, se planifica. ¿Qué tiene que ver la gestión de proyectos con el control de versiones de un software que se está desarrollando?

Es muy recomendable utilizar un control de versiones y, análogamente, una gestión del proyecto en casos en los que el desarrollo de software comporta un trabajo de muchas horas, de muchos archivos diferentes y (aunque no necesariamente) la presencia de varias personas trabajando sobre el mismo proyecto, como por ejemplo en el desarrollo de un ERP (Enterprise Resource Planning).

Un **sistema de control de versiones** es una herramienta de ayuda al desarrollo de software que irá almacenando, según los parámetros indicados, la situación del código fuente en momentos determinados. Es como una herramienta que va haciendo fotos de forma regular, cada cierto tiempo, sobre el estado del código.

En un entorno donde sólo trabajará un programador, bastará con guardar la información del código cada cierto tiempo o cada vez que él guarde la información, junto con los datos principales, como el número de la versión o la fecha y la hora en que se ha almacenado. En un entorno multiusuario, donde muchos programadores diferentes pueden manipular los archivos y donde, incluso, se podrá ofrecer la posibilidad de modificar a la vez un mismo documento, en estos casos es necesario almacenar mucha más información, como qué usuario ha implementado cambios, las referencias de la máquina desde donde se han realizado los cambios, si se está produciendo algún tipo de conflicto...

Esta funcionalidad no sólo será importante en los casos de desarrollo de software, sino también en muchos otros ámbitos. Un ejemplo real para lo que se acaba de explicar se puede encontrar en la redacción de contenidos de Wikipedia, donde muchos redactores crean contenidos mediante una herramienta que permite el trabajo en equipo y el control de los contenidos creados y colgados. Otro caso puede encontrarse en una gestoría o un bufete de abogados que deben redactar un contrato o un documento en colaboración con uno o varios clientes. En este caso, no será necesario utilizar una herramienta como una wiki, quizá sea suficiente utilizar un buen editor de textos que permita la funcionalidad de gestión de cambios, donde cada vez que un usuario diferente tenga que hacer un cambio, quede indicado en el documento con un color diferente en función del usuario que lo haya creado.

En el desarrollo de software, los sistemas de control de versiones son herramientas que pueden facilitar mucho el trabajo de los programadores y aumentar la productividad de forma considerable, siempre que estas herramientas sean utilizadas de forma correcta.



Algunas funcionalidades de los sistemas de control de versiones pueden ser:

- **Comparar cambios** en los diferentes archivos a lo largo del tiempo, pudiendo ver quién ha modificado por última vez un determinado archivo o parte de código fuente.
- **Reducción de problemas de coordinación** que puede haber entre los distintos programadores. Con los sistemas de control de versiones podrán compartir su trabajo, ofreciendo las últimas versiones del código o de los documentos, y trabajar, incluso, de forma simultánea sin miedo a encontrarse con conflictos como resultado de esta colaboración.
- **Posibilidad de acceder a versiones anteriores** de los documentos o código fuente. De forma programada se podrá automatizar la generación de copias de seguridad o incluso almacenar todo cambio efectuado. En el caso de haberse equivocado de forma puntual, o durante un período largo de tiempo, el programador podrá tener acceso a versiones anteriores del código o deshacer, paso a paso, todo lo desarrollado durante los últimos días.
- **Ver qué programador ha sido el último en modificar** una determinada parte de código que puede estar causando un problema.
- **Acceso al historial** de cambios sobre todos los archivos a medida que avanza el proyecto. También puede servir para el jefe de proyectos o cualquier otra parte interesada (stakeholder), con permisos para acceder a este historial, para ver la evolución del proyecto.
- **Devolver un archivo determinado o todo el proyecto entero a uno o varios estados anteriores.**

Los sistemas de control de versiones ofrecen, además, algunas funcionalidades para poder gestionar un proyecto informático y para poder realizar su seguimiento. Entre estas funcionalidades se pueden encontrar:

- **Control histórico** detallado de cada archivo. Permite almacenar toda la información de lo sucedido en un archivo, como todos los cambios que se han efectuado, por quien, el motivo de los cambios, almacenar todas las versiones que ha habido desde su creación...
- **Control de usuarios** con permisos para acceder a los archivos. Cada usuario tendrá un tipo de acceso determinado a los archivos para poder consultarlos o modificarlos o, incluso, borrarlos o crear otros nuevos. Este control debe ser gestionado por la herramienta de control de versiones, almacenando todos los usuarios posibles y permisos que tienen asignados.
- **Creación de ramas de un mismo proyecto:** en el desarrollo de un proyecto hay momentos en los que es necesario ramificarlo, es decir, a partir de un determinado momento, de un determinado punto, es necesario crear dos ramas del proyecto que se podrán seguir desarrollando por separado. Este caso puede darse en el momento de finalizar una primera versión de una aplicación que se entrega a los clientes, pero que es necesario continuar evolucionando.
- **Fusionar dos versiones de un mismo archivo:** permitiendo fusionarlas, cogiendo, de cada parte del archivo, el código que más interese a los desarrolladores. Esta funcionalidad deberá validarse manualmente por parte de una persona.



Componentes de un Sistema de Control de Versiones

Un sistema de control de versiones se compondrá de varios elementos o componentes que utilizan una terminología un poquito específica. A continuación, quedan definidos algunos de estos elementos:

- **Repositorio:** conjunto de datos almacenados, también referido a versiones o copias de seguridad. Es el sitio donde estos datos quedan almacenados. Se podrá referir a muchas versiones de un único proyecto o varios proyectos.

- **Módulo:** se refiere a una carpeta o directorio específico del repositorio. Un módulo podrá hacer referencia a todo el proyecto entero o sólo a una parte del proyecto, es decir, a un conjunto de archivos.

- **Revisión:** en cada una de las versiones almacenadas de un determinado proyecto; podrá hacer referencia a todos los archivos de un proyecto que componen una versión determinada en un instante de tiempo los archivos que han sido modificados desde la última versión.

- **Etiqueta:** información que se añadirá a un módulo para indicar alguna característica específica que le hace especial. Esta información será textual y, en muchas ocasiones, se hará de forma manual. También podrá utilizarse para localizar un conjunto de archivos que componen una revisión completa y que se quieren tener localizados o para poder referenciarlo desde otra versión.

- **Rama:** se utilizará para llevar a cabo un análisis de las revisiones almacenadas sin afectar a las revisiones en curso. Para conseguir esto, es necesario realizar una copia de un módulo o una revisión entera para poder trabajar en paralelo. A esta copia se le llamará rama.

- **Directorio de trabajo:** el programador trabajará sobre este directorio a partir de una copia que habrá realizado del repositorio en su ordenador local.

Clasificación de los Sistemas de Control de Versiones

Se puede generalizar la estructura de las herramientas de control de versiones teniendo en cuenta la clasificación de los sistemas de control de versiones, que pueden ser locales, centralizados o distribuidos.

Sistemas Locales

Los sistemas de control de versiones locales son sistemas que permiten llevar a cabo las acciones necesarias de forma local. Si no se utiliza ningún sistema de control de versiones concreto, un programador que trabaje en su propio ordenador podrá ir haciendo copias de seguridad, de vez en cuando, los archivos o de las carpetas con las que trabaje. Este sistema implica dos características específicas:



- El mismo programador será la persona que deberá acordarse de ir haciendo las copias de seguridad cada cierto tiempo, lo que él haya establecido.

- El propio programador deberá decidir dónde llevará a cabo estas copias de seguridad, muy probablemente en local, en otra ubicación del disco duro interno, o en un dispositivo de almacenamiento externo.

Este sistema tiene un alto riesgo porque es un sistema altamente dependiente del programador, pero es un sistema extremadamente sencillo de planificar y ejecutar. Lo habitual, en estos casos, es crear copias de seguridad (que se pueden considerar versiones del proyecto); los archivos se almacenan en carpetas que suelen tener un nombre representativo, como por ejemplo: la fecha y la hora en que se efectúa la copia. Pero, tal y como se ha comentado anteriormente, se trata de un sistema propenso a errores por diversos motivos, tales como que se produzcan olvidos en la realización de la copia de seguridad, o confusiones que lleven al programador a continuar el desarrollo en una de las copias del código, e ir avanzando con el proyecto en diferentes ubicaciones distintos días.

Para hacer frente a estos problemas, aparecieron los primeros repositorios de versiones que contenían una pequeña base de datos en la que se podían registrar todos los cambios efectuados, sobre qué archivos, quién los había hecho, cuándo... Efectuando las copias de seguridad de forma automatizada.

En la figura.1 se puede observar la representación del sistema local, con los archivos y sus copias o versiones.

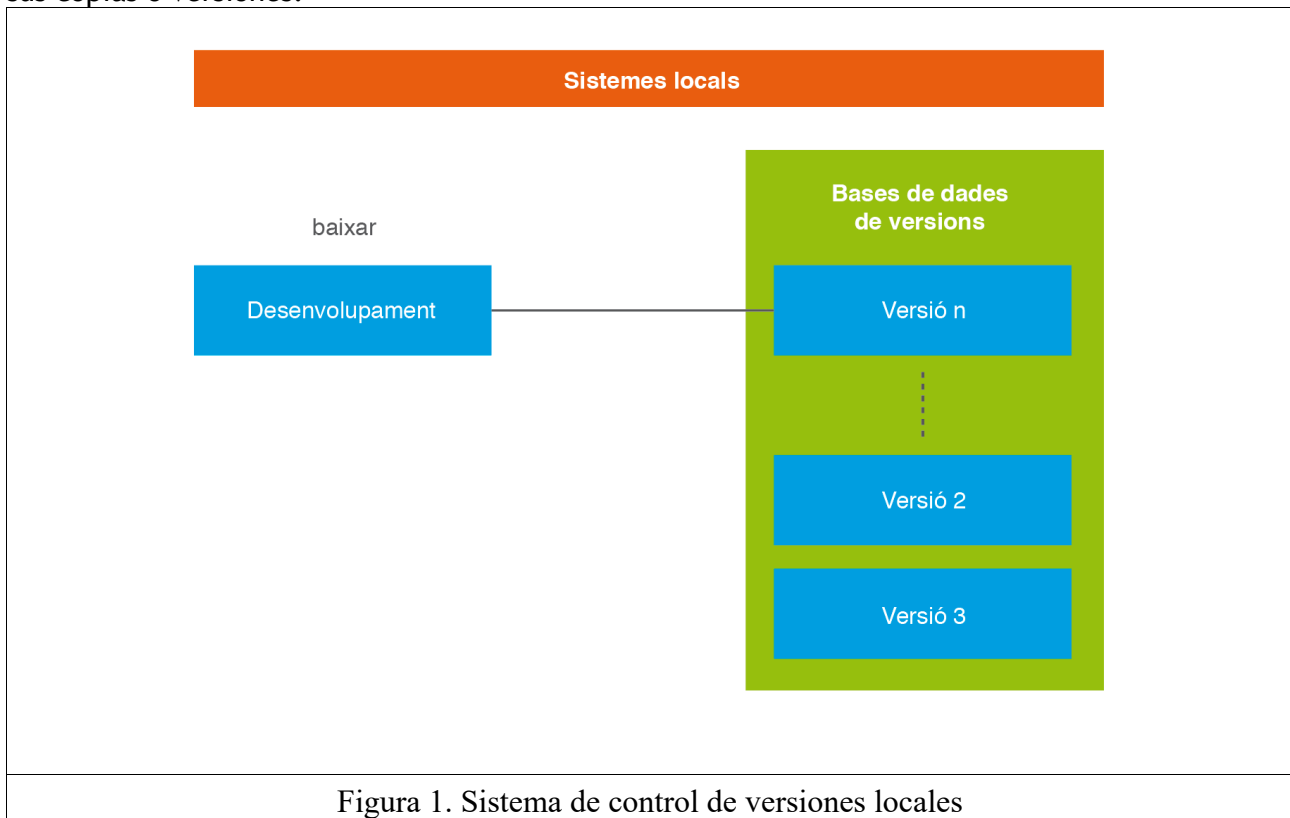
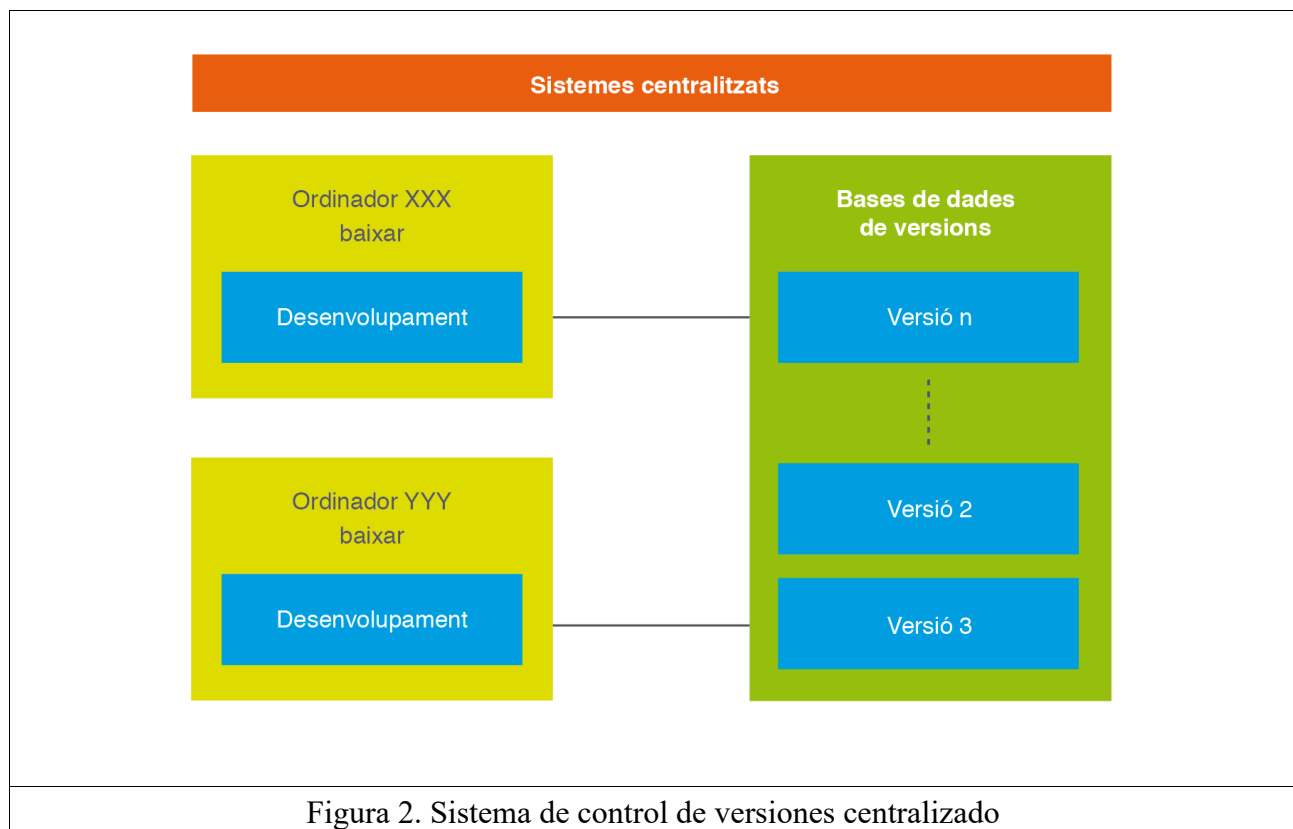


Figura 1. Sistema de control de versiones locales

Sistemas Centralizados

En los sistemas centralizados habrá más de un programador desarrollando un proyecto en más de un ordenador. Desde los diversos ordenadores se podrá acceder a los mismos archivos de trabajo y sobre las mismas versiones almacenadas. Esta es una situación más próxima a las situaciones actuales reales en el desarrollo de proyectos informáticos.

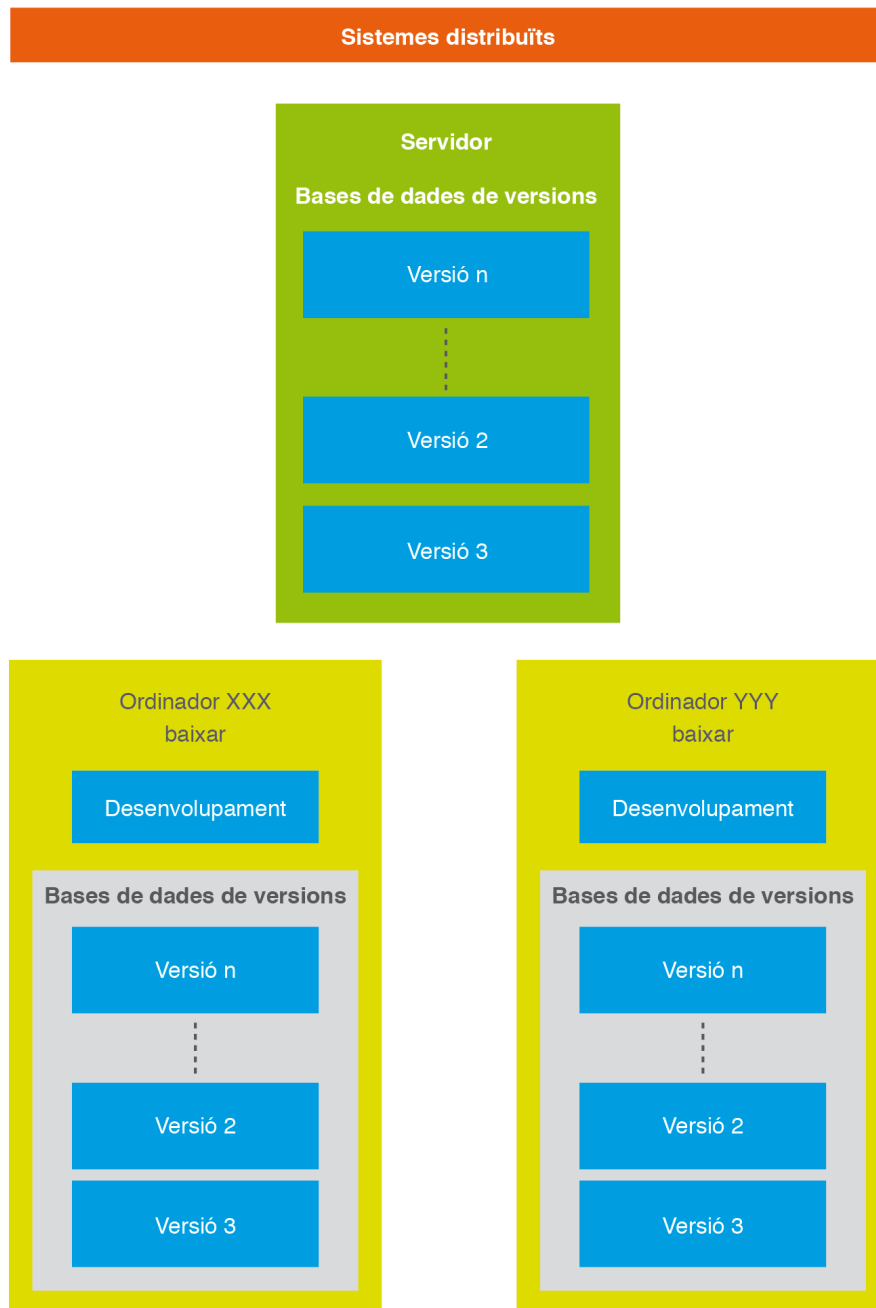
El sistema de control de versiones centralizado es un sistema donde las copias de seguridad o versiones almacenadas se encuentran de forma centralizada en un servidor que será accesible desde cualquier ordenador que trabaje en el proyecto. En la figura.2 se puede observar cómo se representa este sistema de control de versiones centralizado.



Pero este sistema también tendrá sus inconvenientes, a veces muy difíciles de salvar. ¿Qué ocurre si falla el servidor central? Si es un fallo temporal no se podrá acceder durante un pequeño periodo tiempo. Si es un fallo del sistema, habrá que tener preparado un sistema de recuperación o un sistema alternativo (una copia). Durante este tiempo, los diferentes desarrolladores no podrán trabajar de forma colaborativa ni acceder a las versiones almacenadas ni almacenar otras nuevas. Esta es una desventaja difícil de contrarrestar.



Sistemas Distribuidos



Los sistemas de control de versiones distribuidos ofrecen una solución a esta desventaja ofrecida por los sistemas de control de versiones centralizados. Como se puede ver en la figura.3, la solución que ofrecen los sistemas distribuidos es disponer en cada ordenador de trabajo, así como en el servidor, de una copia de las versiones almacenadas. Esta duplicidad de las versiones ofrece una disponibilidad que disminuye muchísimo las posibilidades de no tener accesibilidad a los archivos y en sus versiones.



¿Cuál es la forma de trabajar de este sistema? Cada vez que un ordenador cliente accede al servidor para tener acceso a una versión anterior, no sólo copian ese archivo, sino que hacen una descarga completa de los archivos almacenados.

Si el servidor tiene un fallo del sistema, el resto de ordenadores clientes podrán continuar trabajando y cualquiera de los ordenadores clientes podrá efectuar una copia entera de todo el sistema de versiones hacia el servidor para restaurarlo. La idea es que el servidor es el que gestiona el sistema de versiones, pero en caso de necesidad puede acceder a las copias locales que se encuentran en los ordenadores clientes.

Operaciones básicas de un sistema de control de versiones

Entre las operaciones más habituales de un sistema de control de versiones (tanto en los sistemas centralizados como en los distribuidos) se pueden diferenciar dos tipos: aquellas que permiten la entrada de datos al repositorio y aquellas que permiten obtener datos del repositorio. En la figura.4 se muestra un resumen de estas operaciones.

Entre las operaciones de introducción de datos en el repositorio se pueden encontrar:

- **Importación de datos:** esta operación permite llevar a cabo la primera copia de seguridad o versionado de los archivos con los que se trabajará en local hacia el repositorio.
- **Subir (commit):** esta operación permite enviar, en el repositorio, los datos correspondientes a los cambios que se han producido en el servidor local. No se hará una copia entera de toda la información, sino que sólo se trabajará con los archivos que se hayan modificado en el último espacio de tiempo.

Entre las operaciones de exportación de datos del repositorio se pueden encontrar:

- **Bajar (check-out):** con esta operación se podrá tener acceso y descargar una versión de trabajo de un archivo, un módulo o una revisión a un ordenador cliente.
- **Actualización (update):** esta operación permite realizar una copia de seguridad de todos los datos del repositorio en el ordenador cliente con el que trabajará el programador. Será una operación que se podrá efectuar de forma manual, cuando el programador lo estime oportuno, o de forma automática, como en los sistemas distribuidos, donde cada vez que un cliente accede al repositorio se hace una copia completa en local.

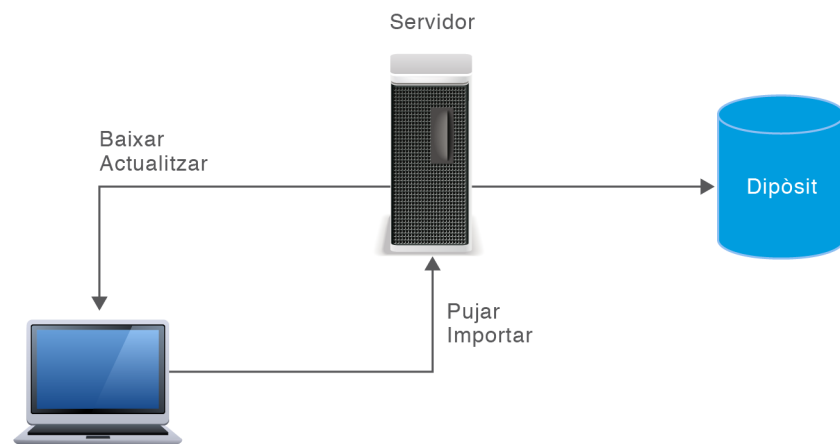


Figura 4. Operaciones que se pueden hacer en un sistema de control de versiones