

CONTROL DE VERSIONES

CODE UD5 – Tema 2

IES Plurilingüe Antón Losada Diéguez

Adrián Fernández González



Tabla de contenido

1. Introducción.....	2
2. Sistema de control de versiones	2
3. Terminología común	2
3.1. Repositorio.....	2
3.2. Revisión.....	2
3.3. Línea base o tronco. Baseline / trunk.....	2
3.4. Rama. Branch	2
3.5. Desplegar. Checkout.....	3
3.6. Publicar. Commit.....	3
3.7. Conflicto.....	3
3.8. Cambio y lista de cambios	3
3.9. Importar y exportar	3
3.10. Integración o fusión. Merge.....	4
3.11. Actualización o sincronizar. Sync / update.	4
3.12. Congelar.....	4
3.13. Revertir. Rollback / revert.....	4

Control de versiones

1. Introducción

El control de versiones es el proceso de gestión de los cambios que se realizan en un sistema, producto o servicio. Cada versión representa el estado del sistema en un momento dado, el estado de todos y cada uno de sus archivos.

2. Sistema de control de versiones

Un sistema de control de versiones es un software o servicio que ayuda a controlar y gestionar los cambios, llevando un registro de las modificaciones realizadas en cada parte del sistema a lo largo del tiempo, permitiendo incluso su vuelta atrás, copia o eliminación.

Existen múltiples sistemas para el control de versiones de software, cada uno con sus utilidades propias, tipos de licencia y uso. Algunos de los más conocidos son Git, Mercurial y Subversion.

3. Terminología común

Aunque cada sistema tiene sus características propias, todos comparten una terminología, unas ideas bases.

Algunos nombres pueden diferir de un sistema a otro.

3.1. Repositorio

El repositorio es el lugar en el que se encuentran los archivos y los metadatos del sistema de control de versiones.

3.2. Revisión

Es la versión del sistema en un momento dado, actual o anterior. Dependiendo del sistema de control de versiones puede ser un número una combinación de números y letras o un código de cifrado.

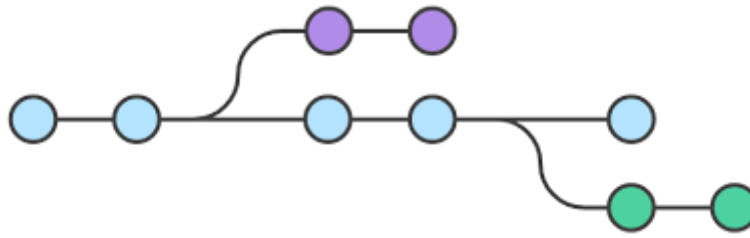
3.3. Línea base o tronco. Baseline / trunk

Es una versión aprobada de un sistema o la versión principal del mismo.

3.4. Rama. Branch

Es cada una de las líneas de trabajo de un sistema.

Puede haber múltiples ramas, que lleven trayectorias distintas, ritmos independientes y hasta equipos separados.



En proyectos libres y/o colaborativos, puede suceder que parte del equipo decida ir por otro lado y crear su propia versión, creando una rama independiente.

3.5. Desplegar. Checkout

Realizar un despliegue o un *checkout* es descargar lo que se encuentra en el repositorio en el sistema local para trabajar con él.

Esto es lo que se realiza cuando se importa en el IDE.

3.6. Publicar. Commit

Realizar una publicación o un *commit* es la operación inversa a la anterior, se sube al repositorio lo que está en el sistema local.

Esto sube todos los archivos al repositorio, añadiendo los que estaba, actualizando los modificados y borrando los eliminados.

Al realizar esta acción, se incluye una descripción más o menos detallada de los cambios realizados. Realizar **esto es de vital importancia** para que el control de las versiones sea efectivo y tenga sentido.

3.7. Conflicto

Un conflicto es una discrepancia entre dos estados de un archivo, ya sea cuando se realiza un *checkout*, un *commit*, una unión de repositorios, etc. Y alguien, por el medio, ha realizado algún cambio.

Estas discrepancias han de ser manejadas de forma individual, decidiendo la acción a tomar en cada caso.

Si los cambios son muy drásticos, puede ser necesaria la modificación manual del archivo para integrar los distintos cambios.

3.8. Cambio y lista de cambios

Los cambios son todas las acciones realizadas en un sistema entre una versión y otra.

Todos estos cambios quedan registrados en una lista o log, en el que se refleja cada cambio individual de cada fichero, pudiendo ver hasta los que han sido eliminados.

3.9. Importar y exportar

La importación y exportación son un proceso similar a *commit* y *checkout*, con la diferencia de que no se migra el historial de cambios. Esto crea una versión limpia, habitualmente utilizada para lanzar el sistema después de su desarrollo.

La primera vez que se sube al repositorio se realiza un *import*.

3.10. Integración o fusión. Merge

Cuando se realiza el trabajo mediante la división en ramas y se quiere mezclar una con otra, se realiza una integración o fusión.

Esta operación es similar a un *commit*, con la salvedad de que las modificaciones se han realizado sobre una rama distinta y se integran en una principal.

3.11. Actualización o sincronizar. Sync / update.

Cuando se trabaja en local y otra persona actualiza el repositorio, es necesario actualizar/sincronizar los archivos locales con los cambios realizados.

3.12. Congelar

Cuando sea necesario evitar que se realicen más cambios en los archivos, porque se prepara una versión para su lanzamiento, por ejemplo, se congela el repositorio, evitando que cualquiera pueda cambiarlo.

3.13. Revertir. Rollback / revert

En caso de que, tras realizar un *commit*, hubiese cualquier tipo de error en los archivos, la mayoría de los sistemas permiten realizar una regresión, una vuelta a la anterior versión.

Hay ciertos sistemas que permiten, además, volver a otra versión que no sea la última.