

Control de Versiones Utilizando SVN

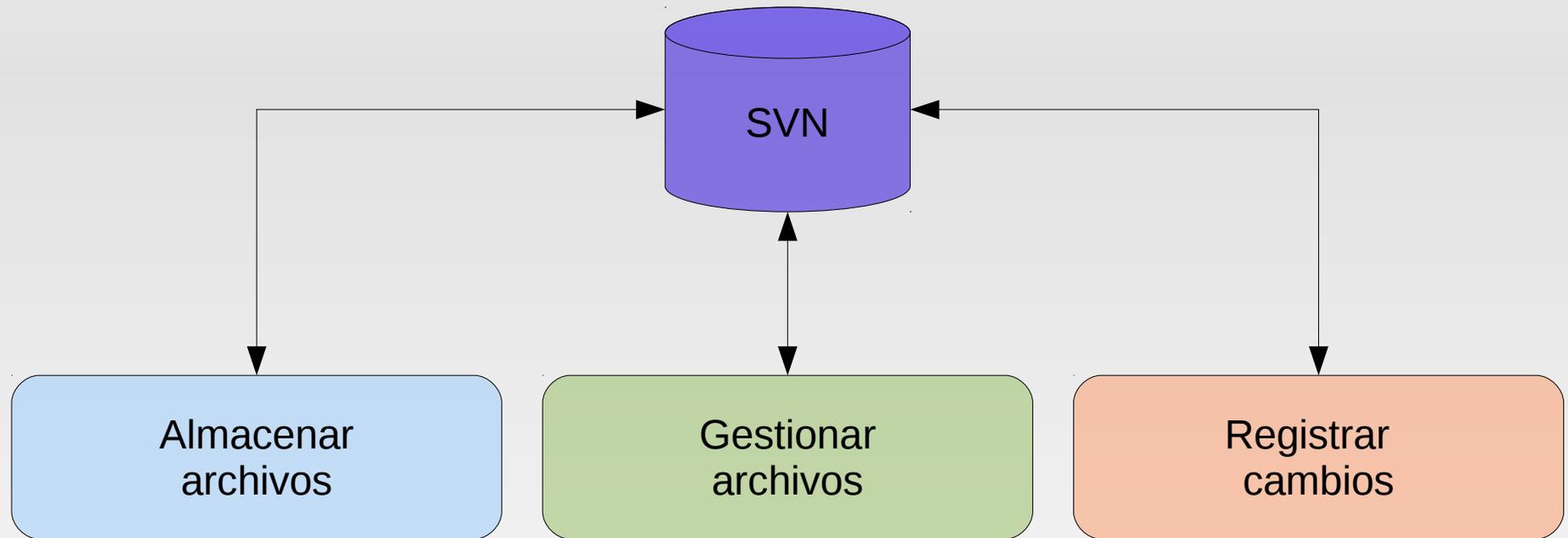


Control de Versiones

*“Se llama **control de versiones** a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo” - San Wikipedia*



¿Qué podemos hacer?



Conceptos Generales

- ★ El elemento principal para el funcionamiento es llamada **repositorio**.
- ★ Se debe trabajar localmente duplicando o descargando la versión del repositorio por medio de **checkout**.

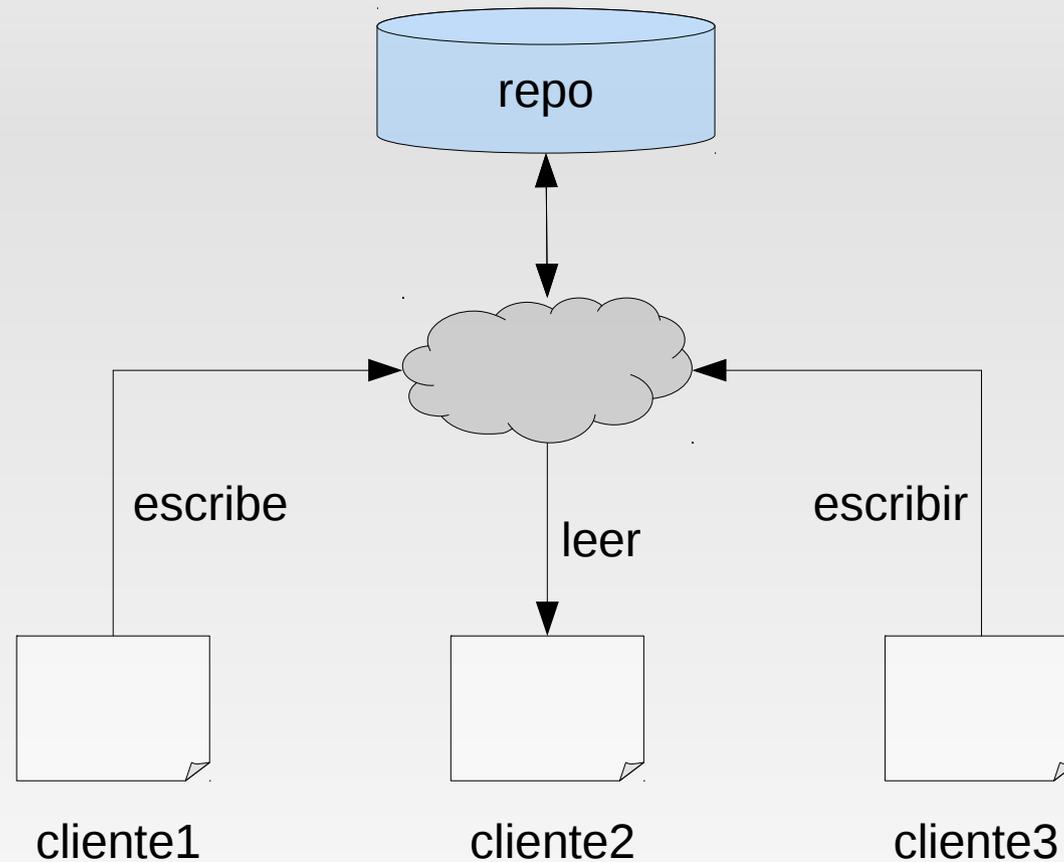


Introducción a Subversion

- ★ **Subversion** es un sistema de control de versiones con licencia OpenSource Apache.
- ★ Utiliza modelo **Cliente-Servidor** para administrar el elemento principal **repositorio**.



Modelo de acciones de trabajo



Principales Acciones (Comandos)

- ★ **Checkout:** descarga una copia privada de la última versión del proyecto, es decir, una working copy, a nuestra máquina.
- ★ **Update:** se usa para actualizar nuestra working copy con los últimos cambios hechos por nuestros compañeros.
- ★ **Commit:** envía al repositorio los cambios que hicimos en nuestra copia privada.

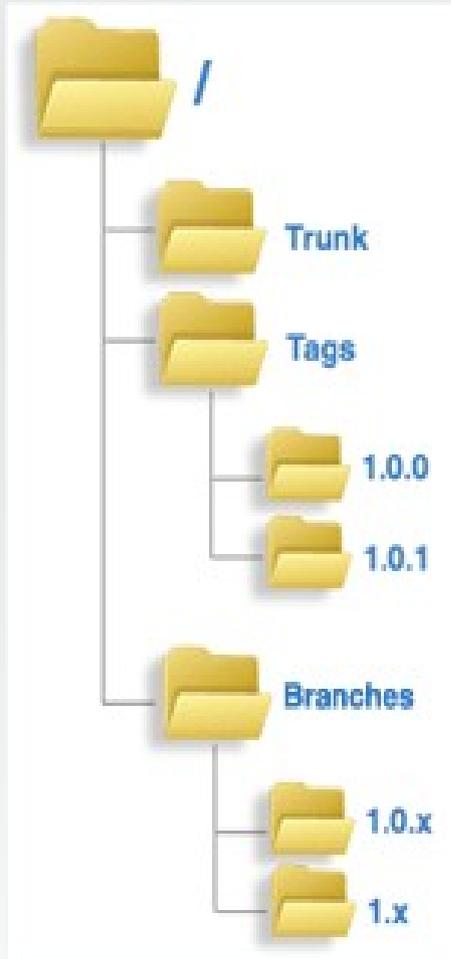


Modelos de Codificación

- ★ Utilizando un **diseño incremental** donde se van agregando módulos al sistema.
- ★ Se utiliza un sistema de árbol para almacenar las fuentes del sistema (sourcecode).
- ★ Rama principal o **trunk** para almacenar el desarrollo principal.
- ★ Los desarrollos en paralelo se guardan en las ramas o **branches**.



Árbol jerárquico



En resumen nuestro árbol de directorios se compone de:

Trunk: Rama de desarrollo principal.

Tags: Rama de gestión de versiones. Reservado para versiones cerradas, por tanto no se desarrollará sobre esta rama.

Branches: Rama con evoluciones paralelas al Trunk.

Cientes y Servidores SVN

- ★ Clientes permiten conectarse al repositorio. En Windows **TortoiseSVN**. *Eclipse* y *Netbeans* pueden instalar plugins.
- ★ Servidores son fácilmente configurables mediante paquetes y configuraciones de **Apache Web Server**.



Introducción uso GoogleCode

- ★ Google Code **provee un hosting** para proyectos (Project Hosting).
- ★ Entrega herramientas como SVN, Wiki y herramientas para administrar repositorios.
- ★ Utiliza cuenta Google (gratuita) para acceder a éstos.



Creación de un Proyecto

Crear



[Create a new project](#)



[Learn more about Project Hosting](#)

Llenar formulario: Elegir Subversion como control de versiones

Usar



wii

test googlecode

Project Home

[Downloads](#)

[Wiki](#)

[Issues](#)

[Source](#)

[Administer](#)



Netbeans & GoogleCode

- ★ El primer paso siempre es hacer **Checkout**. Utilizaremos la carpeta **trunk** del SVN.
- ★ Desde Netbeans **Team > Checkout**
- ★ GoogleCode provee una URI y un nombre de usuario y contraseña:
ej: **<https://tarea1-elo329.googlecode.com/svn/trunk/>**
- ★ Username es el nombre de la cuenta google y la contraseña se obtiene desde un link.



Empezando a trabajar con las versiones

- ★ Para que los cambios se vean reflejados se debe hacer un commit en **Team > Commit.**
- ★ Si trabajamos en conjunto con otros desarrolladores siempre se recomienda antes de hacer nuevos cambios realizar un Update **Team > Update.**
- ★ Podemos revisar las diferencias entre nuestras copias locales y las del repositorio con **Team > Diff**



Buenas prácticas de uso

- ★ Para evitar errores se recomienda:
 - Tener bien definida la estructura del proyecto (trunk ,tags, branches).
 - **Siempre** hacer un Update antes de empezar a trabajar.
 - Realizar commits sólo al final de la jornada o cuando se estime que es necesario reflejar cambios.
 - Realizar commits de archivos relevantes ("de a uno") "don't blanket commit".
 - Agregar pequeños comentarios a los Commits para describir lo que se realizó.



¿Qué rescatamos de Subversion?

- ★ Facilita el trabajo colaborativo.
- ★ Entrega historial de revisiones y copias.
- ★ Puede ser visto en una páginas web y hacer correcciones 'on the fly'.
- ★ Permite manejar varios tipos de archivos.
- ★ Manejo de branches y tags.

Referencias y más info

- ★ Control de Versiones con SubVersion
<http://svnbook.red-bean.com/nightly/es/index.htm>
- ★ Buenas prácticas para el uso de SubVersion
<http://blogs.tecsisa.com/articulos-tecnicos/buenas-practicas-de-gestion-de>
- ★ Getting Started Google Code
<http://code.google.com/p/support/wiki/GettingStarted>
- ★ ¿Cómo usar TortoiseSVN [Windows]?
<http://nosoyfriky.wordpress.com/2010/04/21/como-usar-tortoisesvn-manua>
- ★ The Perfect Subversion Server [Debian/Ubuntu]
<http://www.howtoforge.com/the-perfect-subversion-server-debian-lenny-is>