

## ¿Por qué cambia un producto de software?

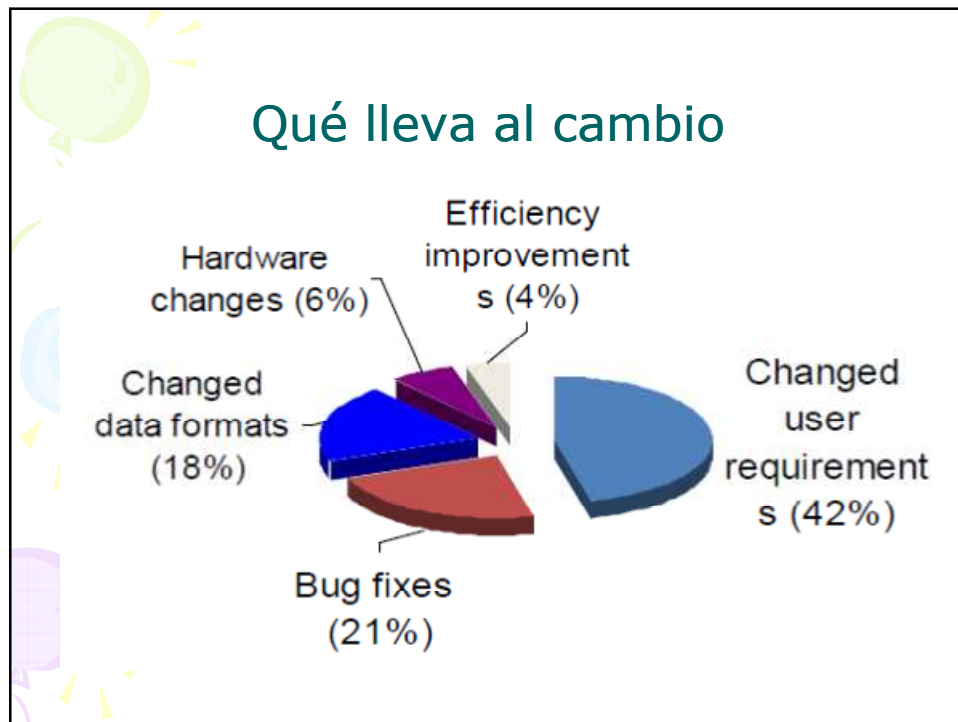
1. Modificación de requerimientos
2. Solución de problemas del producto
3. Cambios en el negocio (costumbres, reglas, necesidades, alcance)
4. Cambios en infraestructura del software (hardware, sistemas operativos, tecnologías, componentes adquiridos)
5. Cambios en el cliente (necesidades, gustos, aprendizaje)
6. Cambios en usuarios (formas de uso, gustos, costumbres, aprendizaje, envejecimiento del software)

3

## Problemas del producto

1. Defectos remanentes
2. Eliminación de detalles molestos
3. Adaptación a nueva infraestructura
4. Mejora de algoritmos

4



### Continuidad

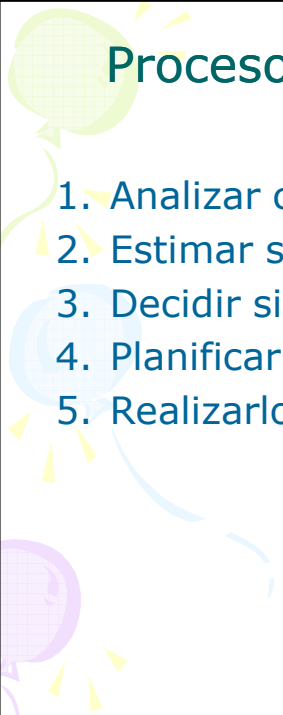
- **El cambio de un producto de software es continuo**
- Relacionado con dos temas:
  - Mantenimiento
    - visto como correcciones y remiendos, reingeniería y otras actividades
  - Evolución
    - visto de manera integral, como un sistema que se va adaptando a su ambiente y debe cambiar

6



## ¿Qué hace el desarrollador?

- Puede rechazar cambios hasta terminar proyecto
  - no funciona en la vida real
- Puede aceptar todo lo que le piden
  - peligroso, nunca termina
- Debe utilizar un procedimiento de control o gestión de cambios
  - solución más racional y
  - necesaria para certificarse en madurez



## Proceso de gestión de cambios (1/4)

1. Analizar cambios propuestos
2. Estimar su impacto y el costo derivado
3. Decidir si es factible
4. Planificarlo
5. Realizarlo

## Proceso de gestión de cambios (2/4)

- Preparar petición de cambio (usuario o cliente)
  - Cambios solicitados
  - Justificación
- Cliente revisa solicitud, evitando duplicidades o cambios inadecuados
  - Acepta o Rechaza, informando al solicitante

## Proceso de gestión de cambios (3/4)

- Se analiza
  - Se revisa impacto (elementos que deberán cambiarse; ver configuración)
  - Se calcula esfuerzo necesario, costo y tiempo necesario
- Un comité decide si se aplica
  - De inmediato
  - En una versión posterior
  - No se considera factible y se rechaza

## Proceso de gestión de cambios (4/4)

- Si se acepta de inmediato, deben:
  - Modificarse la planificación
  - Asignar recursos (responsable)
  - Para cada elemento afectado seguir procedimiento de desarrollo de cambio en la **configuración**
  - Verificar su terminación
  - Integrar en el proyecto (quizá afecte la siguiente **versión**)
  - Avisar a todos los involucrados para desechar versiones anteriores

## Conceptos importantes relacionados

- Línea base
  - Versión oficial, validada por autoridad, de un elemento del proyecto: plan, componente, modelo
- Configuración
  - Conjunto de elementos que integran el software
- Versiones
  - Diferentes formas de un elemento



## Configuración

- Configuración:
  - Disposición de las partes que componen un algo y le dan su forma peculiar
  - Disposición de las partes de un sistema que resuelven un problema
- Configuración del software:
  - Colección de archivos (ejecutables, fuente, figuras, textos, datos), modelos y documentos materiales (manuales)

13



## Configuración vista por usuario

- Un producto de software viene en un CD, DVD o se descarga de Internet y consiste de un instalador y una serie de archivos (a veces enorme) organizados en una jerarquía.

14

## Ejemplo: 7Zip (16 archivos)

Lang	7z
7z.dll 4.65.0.0 7z Standalone Plugin	7z-7-Zip Console Igor Pavlov
7zCon.sfx Archivo SFX 137 KB	7z.sfx Archivo SFX 137 KB
7zG 7-Zip GUI Igor Pavlov	7zFM 7-Zip File Manager Igor Pavlov
7-zip.dll 4.65.0.0 7-Zip Shell Extension	7-zip Archivo de ayuda HTML compil... 87 KB
copying Documento de texto 27 KB	7zip_pad Documento XML 6 KB
History Documento de texto 29 KB	descript.ion Archivo ION 1 KB
readme Documento de texto 2 KB	License Documento de texto 3 KB
	Uninstall


15

## 7Zip, carpeta Lang (74 archivos)

af	es	it	pl	uk
ar	et	ja	ps	uz
ast	eu	ka	pt	va
az	ext	ko	pt-br	vi
ba	fa	ku	ro	zh-cn
be	fi	ku-ckb	ru	zh-tw
bg	fr	lt	si	
bn	fur	lv	sk	
br	fy	mk	sl	
ca	gl	mn	sq	
cs	he	mr	sr-spc	
cy	hr	ms	sr-spl	
da	hu	nb	sv	
de	hy	ne	ta	
el	id	nl	th	
en.ttt	io	nn	tr	
eo	is	pa-in	tt	

16






## Configuración vista por el desarrollador

- Además de los elementos anteriores,
  - Documentos y archivos de requerimientos
  - Modelos de análisis y diseño
  - Diccionario de datos
  - Esquema de base de datos
  - Tablas de datos
  - Programas fuente
  - Archivos auxiliares para construir el producto
  - Casos de prueba y resultados

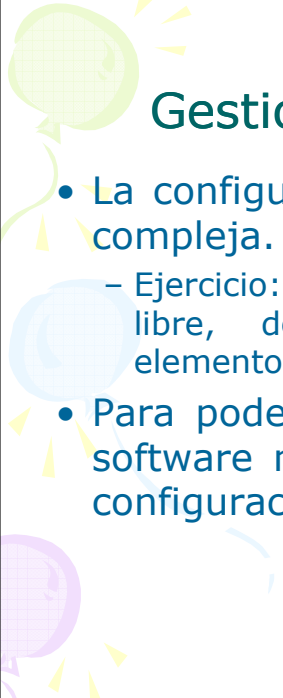
17



## Cambios y configuración

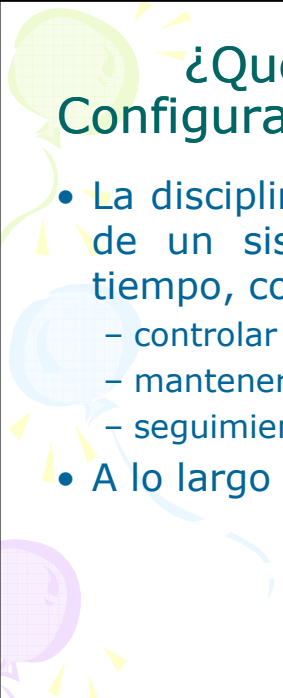
- En cada momento se tiene una configuración
- Cada cambio produce una nueva configuración
- ¿Cómo se controlará la explosión de configuraciones?

18



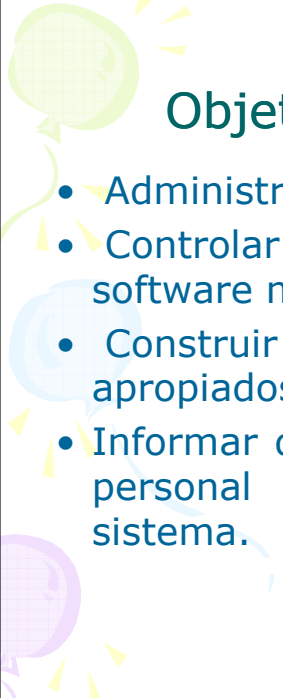
## Gestión de la configuración

- La configuración del software moderno es compleja.
  - Ejercicio: busque un producto de software libre, descárguelo y analice todos los elementos que lo forman
- Para poder administrar un proyecto (y el software mismo) se requiere gestionar la configuración



## ¿Qué es la Gestión de la Configuración del Software (GCS)?

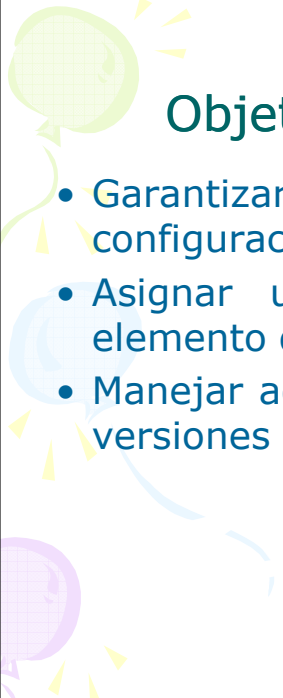
- La disciplina de identificar la configuración de un sistema en puntos discretos del tiempo, con el propósito de:
  - controlar los cambios,
  - mantener la integridad y
  - seguimiento de esta configuración
- A lo largo del ciclo de vida del sistema



## Objetivos de la GCS (1/2)

- Administrar los cambios al sistema.
- Controlar la evolución del producto de software mediante versiones.
- Construir el sistema con los componentes apropiados.
- Informar del estado de la configuración al personal del proyecto y al cliente del sistema.


21



## Objetivos de la GCS (2/2)

- Garantizar que se entregue la configuración correcta al cliente.
- Asignar un nombre adecuado a cada elemento de software que se produzca.
- Manejar adecuadamente las bibliotecas de versiones del software.

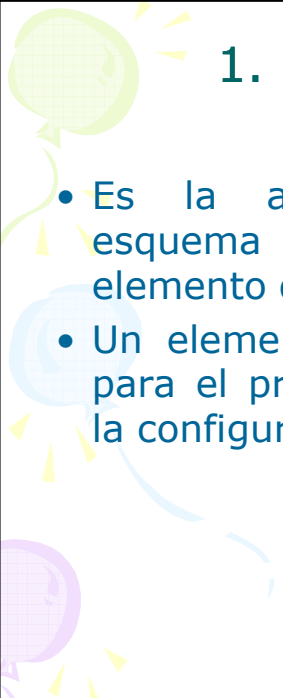
22



## Actividades

1. Identificación de la configuración.
2. Control de la configuración.
3. Informe del estado de la configuración.
4. Auditoría de la configuración.
5. Planear la administración de la configuración.

23



## 1. Identificación de la Configuración

- Es la actividad de documentar un esquema de identificación de cada elemento del sistema.
- Un elemento del sistema es una unidad para el propósito de la administración de la configuración.

24

## Elementos de software sujetos a GCS

- Planes del proyecto
- Especificación de requerimientos de software
- Especificación de diseño
- Código fuente
- Procedimiento de prueba
- Casos de prueba y resultados obtenidos

25

## Elementos de software sujetos a GCS

- Manuales de instalación y de operación
- Descripción de la base de datos
- Manual de usuario final
- Manual de referencia
- Documentos de mantenimiento
- Estándares y procedimientos de la ingeniería de software

26

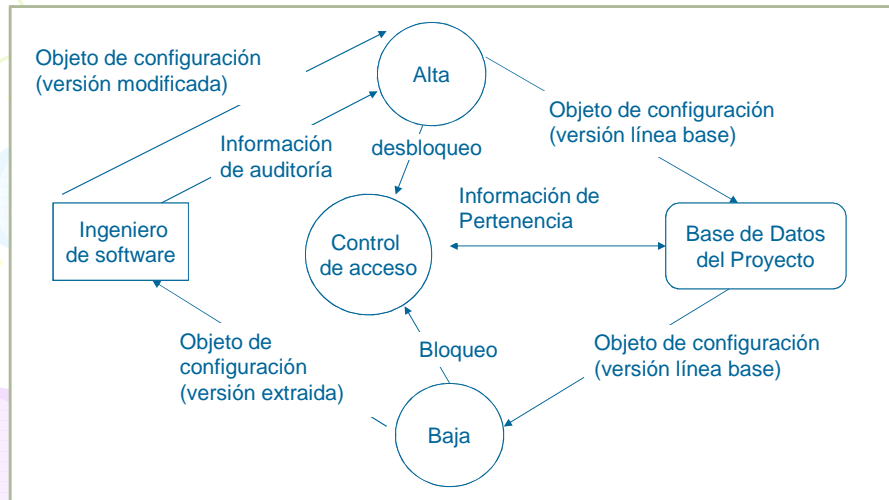
## Identificación

- Cada elemento debe tener una identificación única; no es fácil
- Formas de identificar
  - Nombre y número (Ejemplo: Eclipse 3.4)
  - Nombre y conjunto de atributos. Ejemplo:
    - AC3D (lenguaje= java, plataforma=XP, fecha=Ene2003)
  - El nombre puede incluir varias partes: proyecto, sistema, subsistema, tipo de elemento, nombre particular. Ejemplo largo:
    - org.apache.ant.source\_1.7.1.v20100518-1145
  - El número o atributos identifican una **versión**

## 2. Control de la configuración

- Se controlan los elementos de software
- Se establecen los procedimientos necesarios para proponer, evaluar, revisar, aprobar e implementar los cambios a los elementos en una **línea de base**.
- Se define un procedimiento formal para obtener la autorización para hacer el cambio.

## CONTROL DE ACCESO Y SINCRONIZACIÓN PARA CAMBIOS



29

### 3. Informe del estado de la configuración

- Vehículo por el cual el administrador del proyecto valora los efectos de la GCS.
- Registro y reporte de la información necesaria para manejar una configuración. Incluye:
  1. Lista de la identificación aprobada de la configuración.
  2. Estado de cambios propuestos.
  3. Estado de la implementación de los cambios aprobados.

30

## 4. Auditoría de la configuración

- La auditoría de la configuración es el proceso de verificar y validar el hecho que una configuración propuesta está completa y es consistente. Se divide en:
  1. Auditoría de la configuración física
  2. Auditoría de la configuración funcional

31

## 4. Auditoría de la configuración

1. Auditoría de la configuración física
  - Determina que todos los elementos identificados como parte de la configuración estén presentes y en la línea base del producto.
2. Auditoría de la configuración funcional
  - Cada elemento de la configuración se desempeña de acuerdo con lo establecido en su especificación de requerimientos de software.

32



## 5. Planear la administración de la configuración

- Planear la administración de la configuración significa documentar los métodos a ser usados para identificar los elementos de software, controlar e implementar los cambios, registrar y reportar el estado y las auditorías de la configuración

33

## Puntos del plan.

- Los puntos del documento que contiene el Plan de GCS son:
  1. Introducción.
  2. Administración.
  3. Actividades a realizar de la GCS.
  4. Herramientas, Técnicas y Metodología.
  5. Registro de la configuración.
  6. Auditoría de la configuración.

34

## Ejemplo del contenido del Plan de configuración:

### 1. Introducción

- Proyecto
- Fases
- Objetivos y riesgos del plan de GCS
- Alcance
- Recursos

35

## Ejemplo del contenido del Plan de configuración:

### 2. Actividades

- Identificación de la configuración
  - Definición de líneas base
  - Componentes
  - Nombres
  - Dirección de la Biblioteca
- Control de la configuración. Procedimientos de:
  - Solicitud de cambio, evaluación del cambio, aprobación del cambio e implementación del cambio
- Informe del estado de la configuración
- Auditorías a la configuración

36

## Ejemplo del contenido del Plan de configuración:

3. Calendarización de las actividades
4. Recursos para la GCS
5. Mantenimiento del plan.
  - Responsable del seguimiento del plan
  - Responsable de la actualización del plan

37

## Versiones y desarrollo

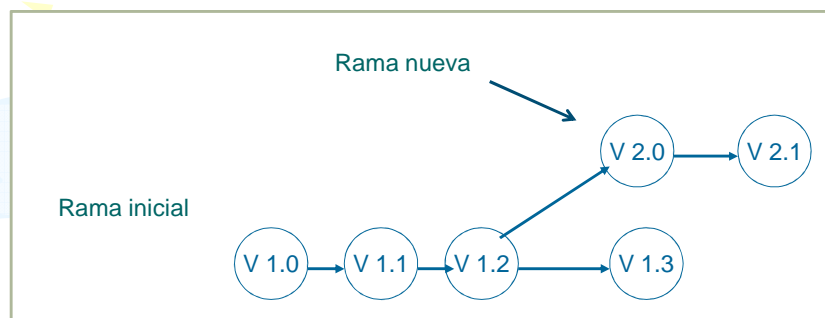
- En sistemas antiguos, había una "liberación"
  - a partir de ahí, había versiones como mantenimiento
- En sistemas modernos, iterativos,
  - cada iteración genera una versión liberable y quizá varias construcciones internas
  - el cambio y las versiones comienzan desde el inicio del proyecto

38

## Versiones

- Cada elemento de la configuración va cambiando.
- Debe distinguirse la versión actual de las anteriores
- En ocasiones se emplean versiones anteriores
- A veces hay cambios opcionales que generan ramas (diferente sistema operativo, 32/64 bits, distinto hardware)

## Control de versiones grafo de evolución

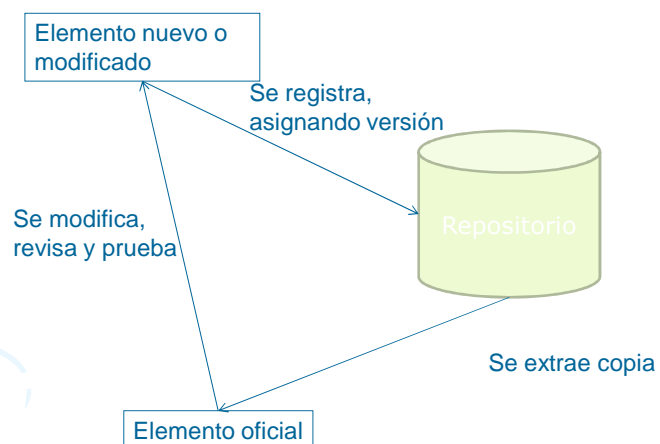



Las versiones pueden ser numéricas o por atributos (fecha, plataforma, cliente)

## Software para control de versiones

- Control de versiones a pié es molesto e inseguro; propenso a olvidos
  - Ejemplos: empaclar versiones, uso de diferente carpeta, salvar en CD ó dispositivo USB
- Debe ser forma automática, que impida cambios arbitrarios y evite olvidos

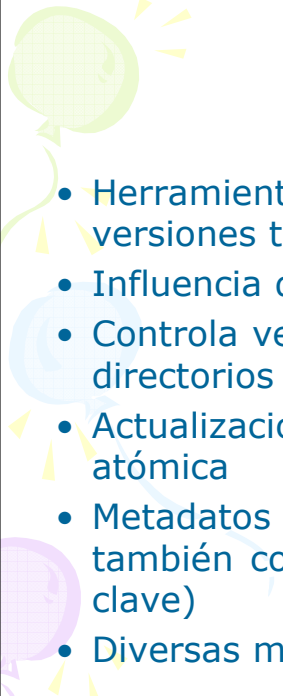
## Software para control de versiones





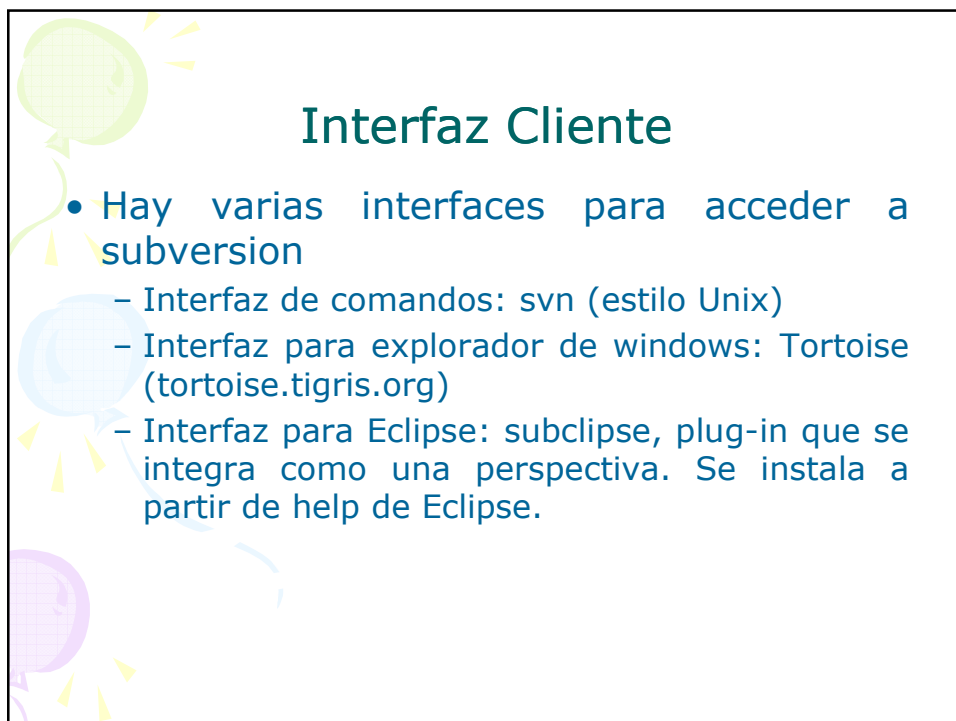
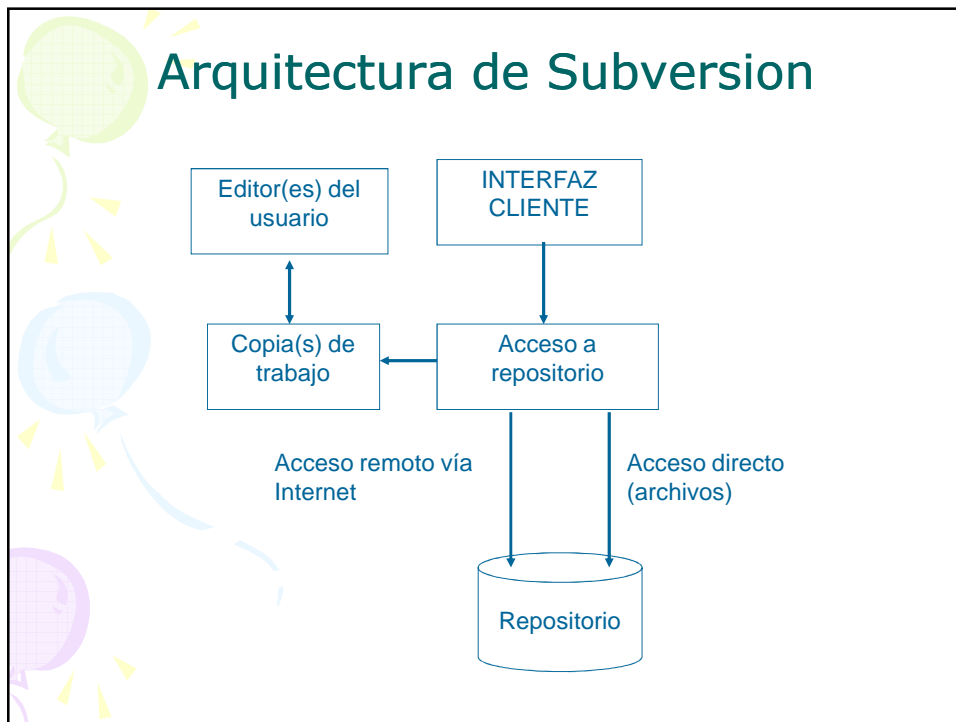
## Algunas herramientas

- CVS
- SVN (Subversion)
- Bazaar
- GIT



## Subversion

- Herramienta de software libre para control de versiones [tigris.org](http://tigris.org)
- Influencia de CVS (no se tratará)
- Controla versiones de elementos simples y de directorios enteros
- Actualización de repositorio de manera atómica
- Metadatos asociados a los elementos, también con versión (propiedades y palabras clave)
- Diversas maneras de acceder al repositorio





## Interfaz svn

- Para algunas cosas es útil el manejo de comandos
- Creación de repositorio:
  - ...>svn create path del folder para el repositorio
- Debe ser un folder compartido, que se manejará por el nombre que se le dio.
- Puede usarse uno prestado, como
  - <http://svn.sourceforge.net/svnroot/gallery>
  - <http://subclipse.tigris.org/>



## Acceso al repositorio

- Existen varias formas de acceso: como archivo o remoto.
  - Siempre se indica como URL:
  - <file:///repositorio>
  - <http://sitio/repositorio>
  - <file://sitio/repositorio>
  - https, ssh, svn, svn+ssh
- Requieren adaptador cuando se usa Java





## Tortoise Preparación y consulta

- Instalación:
  - Se baja paquete y se autoinstala
  - <http://tortoisesvn.net/download.html>
  - queda ligado al Explorador de Windows, usando **Botón Derecho**
- Creación de repositorios:
  - En un folder (vacío),
    - BD->"Tortoise-SVN"->"Create Repository Here"

Nota: BD="clic en Botón Derecho"



## Tortoise carga inicial y consulta

- Registrar un folder con su contenido:
  - BD->"Tortoise-SVN" -> Import.
- Consultar lo que se tiene:
  - BD->"Tortoise-SVN"->"Repo Browser";
  - elegir repositorio.
- Árbol de versiones:
  - BD->"Tortoise-SVN"->Repo Graph

Nota: BD="clic en Botón derecho"



## Tortoise carga inicial y consulta

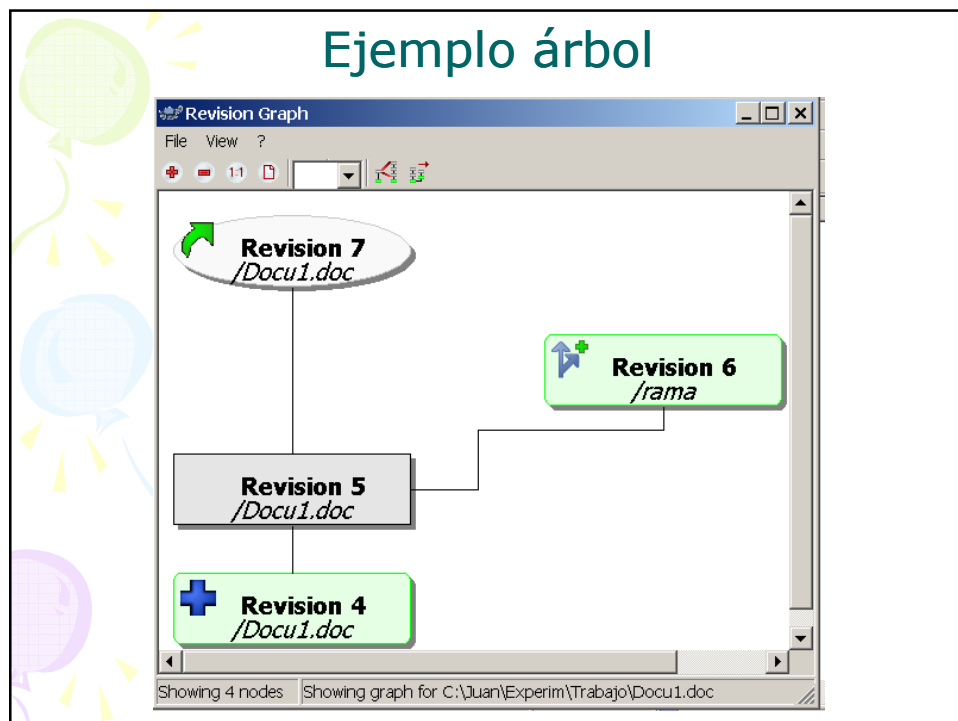
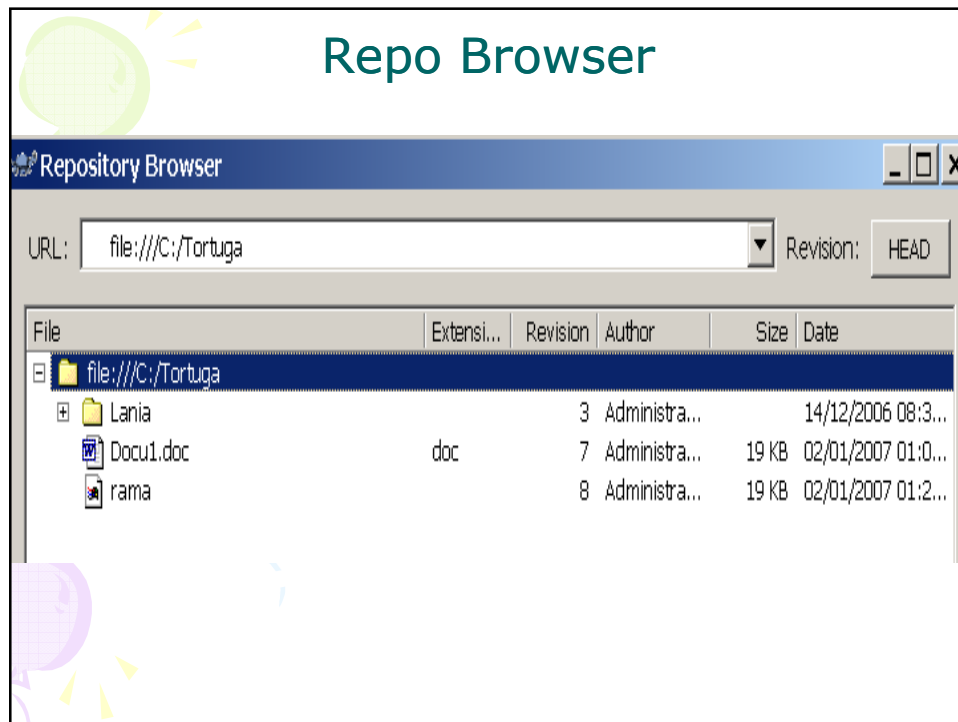
- Alternativa mejor para registrar archivos:
  - Abrir el repositorio (repo browser)
  - Dentro de éste, crear una nueva carpeta para cada proyecto que se desea controlar
  - Posicionados sobre una carpeta donde se trabajará, hacer tortoise->checkout de la carpeta creada en el repositorio
    - aparecerá la carpeta con un recuadro verde y una marca (✓)



## Tortoise carga inicial y consulta

- Arrastrar los archivos que se desean guardar (evite los de tipo binario, como gif, jpg y similares) a la carpeta extraída
- Posicionados sobre la carpeta extraída, dar:
  - BD en TortoiseSVN->add
  - añadir los archivos que se acaban de arrastrar
- En la misma carpeta,
  - BD->Tortoise Commit
  - para que los guarde en la carpeta controlada en el repositorio

Nota: BD="clic en Botón derecho"

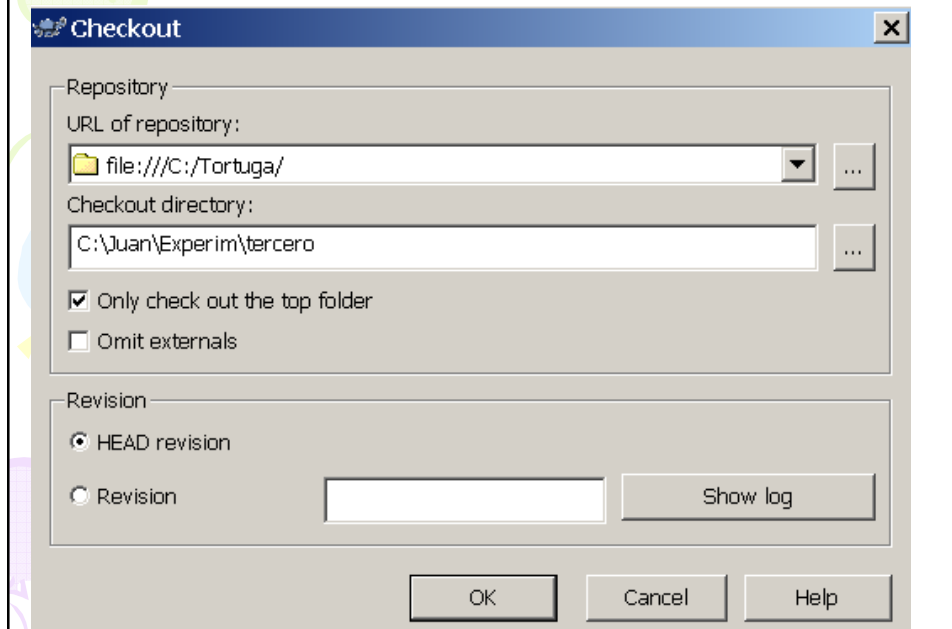


## Tortoise Operación diaria (1)

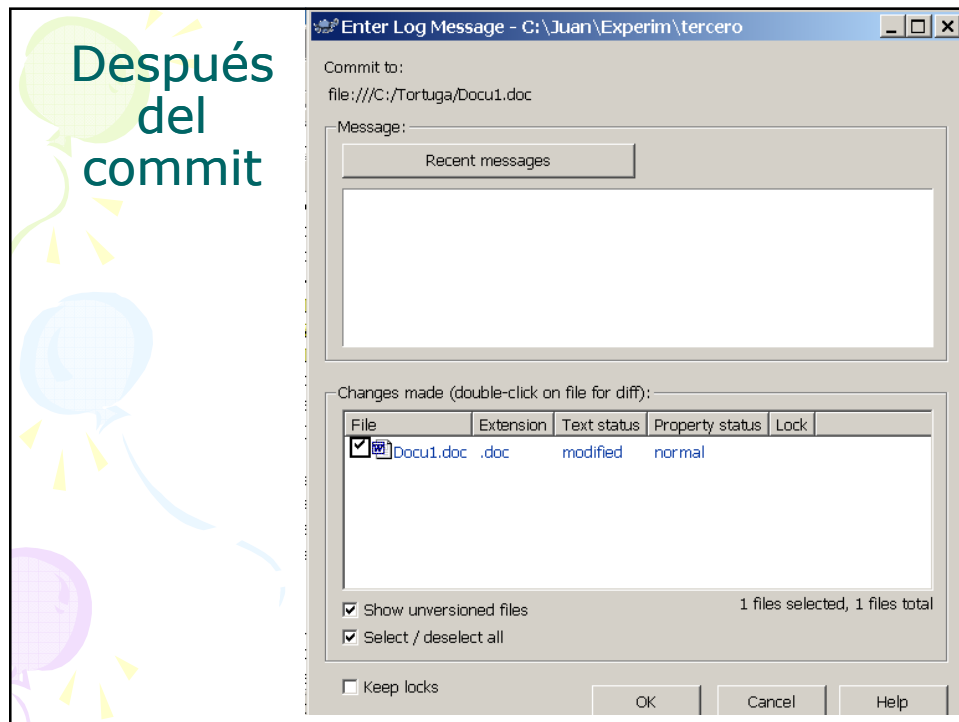
- Extraer copia:
  - En directorio de trabajo,
    - BD->"SVN Checkout",
  - saca del repositorio lo que se desea.
  - última versión: HEAD; otras, modificar marca.
- Actualizar versión que se trabaja:
  - BD->"SNV Update"
- Guardar nueva versión:
  - sobre el archivo,
    - BD->"SVN commit".
  - Si se hace sobre folder, sólo actualiza los que cambiaron.

**Nota: BD="clic en Botón derecho"**

## Checkout







## Tortoise Operación diaria (2)

- Comparar contra versión oficial:  
– BD->SVN Diff.
- Comparar con otras versiones concurrentes:  
– BD->“SVN Check for modifications”
- Solución de conflictos:  
– BD->“Edit conflicts”

Nota: BD=“clic en Botón derecho”

## Tortoise otros

- Ramificaciones y marcas:

- en directorio, en vez de checkout se hace branch/tag,

- creará una copia de la última versión guardada, con otro nombre;

- debe quedar en directorio diferente o con nombre diferente en el repositorio.