

---

# Sistemas de Control de Versiones CVS y Subversion

*Pablo Barrera González*

*barrera@gsync.escet.urjc.es*



## ¿Qué es un sistema de control de versiones?

- Sistema para mantener un seguimiento durante un desarrollo
- Se basa en las revisiones incrementales de los archivos
- Permite obtener cualquier revisión/versión en cualquier momento
- Permite trabajar de forma concurrente a varios desarrolladores
- Añade información y comentarios al efectuar cambios
- Es independiente del contenido
- Puede funcionar en red o localmente
- Es útil tanto en desarrollos pequeños (de 1 desarrollador) como en grandes (de varios)

## ¿Qué no es?

- No es un compilador (gcc, g++, javac, mcs, etc)
- No construye aplicaciones (make, ant, jam, etc)
- No elimina el coordinador del proyecto (el jefe)
- No elimina la comunicación entre usuarios (mail, gforge)
- No lleva control de bugs (bugzilla)
- No es un depurador (gdb, ddd)
- No prueba aplicaciones (Unit)

---

## Partes básicas (Vocabulario)

**Repositorio** Lugar de almacenamiento de los datos de uno o varios proyectos. Es un directorio en alguna máquina (por ejemplo: /var/lib/cvs)

**Módulo** Es un directorio específico del repositorio. Puede identificar una parte del proyecto o ser un proyecto en sí.

**Revisión** Cada una de las versiones parciales o cambios en los archivos o repositorio completo. La evolución del sistema de versiones se mide en revisiones. Cada cambio se considera incremental.

**Etiqueta** Información textual que se añade a un conjunto de archivos (o a un módulo completo) para indicar algún hito importante (p.e.: VERSION\_0\_1).

**Rama** Revisiones paralelas de un módulo para efectuar cambios sin tocar la evolución principal. Se suele emplear para pruebas o para mantener cambios en versiones viejas.

## Operaciones básicas

- Introducir datos en el repositorio:

**Import** Enviar la primera copia de un módulo. Sólo se usa una vez.

**Commit** Mandar los cambios locales al repositorio.

- Sacar datos del repositorio:

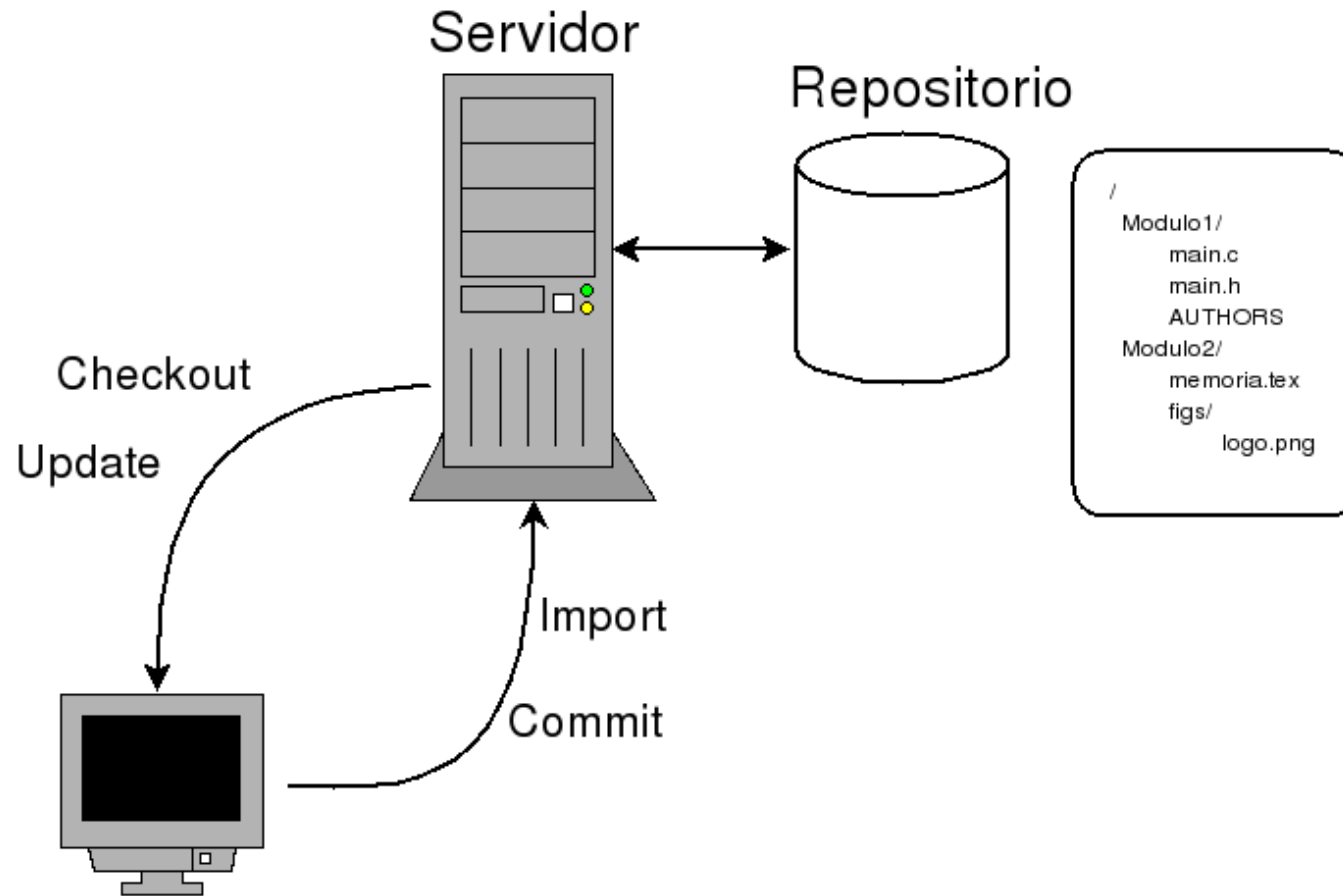
**Check-out** Descargar una versión de trabajo a tu máquina. Sólo se hace una vez.

**Update** Actualizar en la copia local los cambios del repositorio.

## Comentarios

- Cada operación debe comentarse
- Indica qué problema resuelve o qué característica añade
- Sirven para informar al resto de usuarios y para el seguimiento de las versiones

# Escenario



# Subversion

---

# Subversion

- Subversion es un sistema de control de versiones
- Muy parecido al conocido CVS
  - Aparece para suplir las limitaciones de éste
- Está empezando a ser adoptado por muchos desarrollos libres
- Mantiene un repositorio centralizado
- Permite múltiples desarrolladores
- Es independiente del contenido
- Admite archivos binarios sin problemas

---

## Características

- Las revisiones se hacen sobre todo el árbol de directorios
  - Varios cambios (cambiar/añadir/borrar) pueden ir dentro de la misma revisión
- Las diferencias almacenadas son binarias. El tratamiento de los archivos binarios es igual que el de los archivos de texto.
- La copia/movimiento de archivos no es una operación costosa.

## Tipos de servidores

- Los servidores están identificados por una URL
- El funcionamiento local o por red es idéntico: **solo cambia la URL**
- Protocolos soportados
  - file:/// direct repository access (on local disk)
  - http:// access via WebDAV protocol to Subversion-aware Apache server
  - https:// same as http://, but with SSL encryption.
  - svn:// access via custom protocol to an svnserve server
  - svn+ssh:// same as svn://, but through an SSH tunnel.

## Estructura de los módulos

- Para simplificar la utilización de etiquetas y ramas, los módulos suelen tener tres directorios básicos:

**trunk** Directorio para el código fuente

**tags** Directorio para las etiquetas

**branches** Directorio para las ramas

## Crear un repositorio

```
$ svnadmin create /tmp/svn
```

- Debe el fuente no tiene estructura del módulo, puedes usar `svn mkdir`
- Es combeniente tener un módulo por repositorio (pero no necesario)

## Introducir datos

Opción **import**

```
$ svn import file:///tmp/svn
Adding          trunk
Adding          trunk/hsifilter.i
Adding          trunk/pyhsituner.py
Adding (bin)    trunk/hsituner.png
Adding          trunk/TODO
Adding          trunk/hsituner.glade
Adding          trunk/Makefile
Adding          branches
Adding          tags
```

Committed revision 1.

## Bajar copia de trabajo

Opción **checkout** o **co**

```
$ svn co file:///tmp/svn/trunk svn1
A  svn1/hsifilter.i
A  svn1/pyhsituner.py
A  svn1/hsituner.png
A  svn1/TODO
A  svn1/hsituner.glade
A  svn1/Makefile
Checked out revision 1.
```

## Realizar cambios y subirlos

Puedes

- Hacer cambios
- Añadir archivos (**add**)
- Copiar archivos (**copy**)
- Renombrar archivos (**move**)
- Borrar archivos (**delete**)

Para subir

```
$ svn add otroarchivo.txt
```

```
A      otroarchivo.txt
```

```
$ svn ci
```

```
Sending      hsifilter.i
```

```
Adding      otroarchivo.txt
```

```
Transmitting file data ..
```

```
Committed revision 2.
```

## Actualizar la copia de trabajo

Opción **update**

```
$ svn update
```

```
U  hsifilter.i
```

```
A  otroarchivo.txt
```

```
Updated to revision 2.
```

## Resolución de conflictos

```
$ svn ci
Sending          hsifilter.i
svn: Commit failed (details follow):
svn: Out of date: '/trunk/hsifilter.i' in transaction '7'
$ svn update
C  hsifilter.i
A  otroarchivo.txt
```

Updated to revision 4.

<<<<<< .mine

Esto es un cambio local

=====

Los cambios del servidor

>>>>>> .r4

\$ svn resolved hsifilter.i

Resolved conflicted state of 'hsifilter.i'

## Subidos en las ramas

- Para svn las ramas y los tags son iguales
- Funcionan igual que una copia
- Se copia el directorio **trunk** a **branches/rama** o **tags/etiqueta**
- Si modificas algo en el nuevo directorio será una rama, en caso contrario una etiqueta.

```
$ svn copy file:///tmp/svn/trunk file:///tmp/svn/tags/0.1
Committed revision 4.
$ svnlook tree /tmp/svn/ | grep /
/
trunk/
branches/
tags/
0.1/
```

## Exportar una bola

```
$ svn export file:///tmp/svn/tags/0.1 version-0.1
A version-0.1
A version-0.1/hsifilter.i
A version-0.1/pyhsituner.py
A version-0.1/hsituner.png
A version-0.1/TODO
A version-0.1/hsituner.glade
A version-0.1/Makefile
A version-0.1/otroarchivo.txt
Exported revision 4.
$ tar cfz version-0.1.tgz version-0.1/
```

## keywords

- `$LastChangedDate$` o `$Date$`
- `$LastChangedRevision$` o `$Rev$`
- `$LastChangedBy$` o `$Author$`
- `$HeadURL$`
- `$Id$`

Es necesario activarlas

```
svn propset svn:keywords "Date Author" *
```

## Otras opciones interesante

- status
- list
- log
- blame
- help

## Referencias

- <http://svnbook.red-bean.com>
- [http://www.sindominio.net/quique/Traducciones/subversion\\_personal.html](http://www.sindominio.net/quique/Traducciones/subversion_personal.html)