

# Version control with Subversion

Area de Ingeniería Telemática  
<http://www.tlm.unavarra.es>

Grupo de Redes, Sistemas y Servicios Telemáticos

# Contenido

- Qué es el control de versiones
- Creación de un repositorio con Subversion
- Basic workflow
- Tags and branches
- Otras utilidades

# ¿ Version control ?

- Proyecto con varios participantes
- Asegurarse de que todo el mundo posee la última versión
- Evitar que un participante destruya el trabajo de otro
- Almacenar documentos en un repositorio central
- Todos los participantes tienen acceso al repositorio
- Automatiza:
  - Actualizar copia local con los últimos cambios en el repositorio
  - Introducir los cambios locales en el repositorio
  - Registro histórico de los cambios y las versiones

# Procedimiento

- Creación del repositorio
- Importar una versión inicial al repositorio
- Cada participante obtiene (*checks out*) una copia local del repositorio
- Los participantes hacen cambios en su copia local
- Los participantes introducen sus cambios en el repositorio
- Los participantes actualizan su copia local con los cambios introducidos por otros

# Subversion

- *svn*
- Repositorio:
  - Un directorio en una máquina
  - URLs:
    - file:///path/to/repos
    - http://fry.tlm.unavarra.es/path/to/repos
    - https://fry.tlm.unavarra.es/path/to/repos
    - svn://fry.tlm.unavarra.es/path/to/repos
    - svn+ssh://fry.tlm.unavarra.es/path/to/repos
- Versión global de todo el repositorio
- Soporta binarios
- Con texto es más inteligente en los *update*
- Usado?
  - ASF, KDE, GCC, Python, Samba, Mono, PuTTY, Zope, Plone, Xiph, GnuPG, CUPS...

# Creación del repositorio

# Creación del repositorio

```
$ svnadmin create myrepos
```

- Se pueden crear directorios directamente en el repositorio:

- Estructura típica del repositorio:

```
$ svn mkdir file:///Users/daniel/myrepos/trunk -m  
"Trunk"
```

```
$ svn mkdir file:///Users/daniel/myrepos/tags -m  
"Tags"
```

```
$ svn mkdir file:///Users/daniel/myrepos/branches -m  
"Branches"
```

- Listado:

```
$ svn list file:///Users/daniel/myrepos  
branches/  
tags/  
trunk/
```

- Estructura real:

```
$ ls /Users/daniel/myrepos  
README.txt      dav      format      locks  
conf            db       hooks
```

# Importación inicial

- Si ya se tiene un proyecto que se quiere importar

```
$ ls -l miproyecto/  
total 16  
-rw-r--r--    1 daniel  daniel   31 Jul 18 16:00 global.h  
-rw-r--r--    1 daniel  daniel  133 Jul 18 16:00 main.c
```

- Importación en el trunk

```
$ svn import miproyecto file:///Users/daniel/myrepos/trunk  
Adding          miproyecto/main.c  
Adding          miproyecto/global.h
```

```
Committed revision 4.
```

```
$ svn list file:///Users/daniel/myrepos/trunk  
global.h  
main.c
```



# Ver el contenido en el repositorio

```
$ svn cat file:///Users/daniel/myrepos/trunk/main.c
#include <stdio.h>
#include <stdlib.h>
#include "global.h"
```

```
main()
{
    printf("%s\n\n", TITULO);
    printf("Hola mundo!\n");
    exit(0);
}
```

```
$ svn cat
file:///Users/daniel/myrepos/trunk/global.h
#define TITULO "Mi Hola Mundo"
```

# Basic workflow

# Basic workflow: checking out

- Check-out del repositorio o subdirectorio del mismo:

```
$ svn checkout file:///Users/daniel/myrepos/trunk miproy
A    miproy/main.c
A    miproy/global.h
Checked out revision 4.
```

```
$ ls -al miproy/
total 16
drwxr-xr-x   5 daniel  daniel  170 Jul 18 16:36 .
drwxr-xr-x  22 daniel  daniel  748 Jul 18 16:36 ..
drwxr-xr-x  11 daniel  daniel  374 Jul 18 16:36 .svn
-rw-r--r--   1 daniel  daniel   31 Jul 18 16:36 global.h
-rw-r--r--   1 daniel  daniel  133 Jul 18 16:36 main.c
```

- Directorios *.svn* (No tocar!)
- Remotamente (otro repositorio):

```
$ svn checkout
  svn+ssh://fry.tlm.unavarra.es/home/daniel/elrepos/trunk
  miproy
```

# Basic workflow: info

```
$ svn info
Path: .
URL: file:///Users/daniel/myrepos/trunk
Repository UUID: ef34628f-2741-4cbb-a414-c4fba688b72a
Revision: 4
Node Kind: directory
Schedule: normal
Last Changed Author: daniel
Last Changed Rev: 4
Last Changed Date: 2007-07-18 16:33:53 +0200 (Wed, 18 Jul 2007)
```

# Basic workflow: info

```
$ svn info main.c
Path: main.c
Name: main.c
URL: file:///Users/daniel/myrepos/trunk/main.c
Repository UUID: ef34628f-2741-4cbb-a414-c4fba688b72a
Revision: 4
Node Kind: file
Schedule: normal
Last Changed Author: daniel
Last Changed Rev: 4
Last Changed Date: 2007-07-18 16:33:53 +0200 (Wed, 18 Jul 2007)
Text Last Updated: 2007-07-18 16:36:59 +0200 (Wed, 18 Jul 2007)
Properties Last Updated: 2007-07-18 16:36:59 +0200 (Wed, 18 Jul 2007)
Checksum: cdd803325bb623f79f63942c50fc94a3
```

# Basic workflow: nuevos

- Añadir ficheros o directorios al repositorio

```
$ ls -l
total 24
-rw-r--r--  1 daniel  daniel   46 Jul 18 16:40 Makefile
drwxr-xr-x  3 daniel  daniel  102 Jul 18 16:40 docs
-rw-r--r--  1 daniel  daniel   31 Jul 18 16:36 global.h
-rw-r--r--  1 daniel  daniel  133 Jul 18 16:36 main.c
```

- Añadir fichero

```
$ svn add Makefile
A          Makefile
```

- Añadir directorio (con todo lo que contiene)

```
$ svn add docs
A          docs
A          docs/README.txt
```

- Lo hace sin contactar con el repositorio

# Basic workflow: modificaciones

- Modificamos algún fichero (por ejemplo global.h)
- Podemos ver los cambios

```
$ svn diff global.h
```

```
Index: global.h
```

```
=====
```

```
=====
```

```
--- global.h      (revision 4)
```

```
+++ global.h      (working copy)
```

```
@@ -1,2 @@
```

```
    #define          TITULO "Mi Hola Mundo"
```

```
+#define          DESPEDIDA "Adios"
```

- Lo hace sin contactar con el repositorio (en .svn tiene una copia de lo que se bajó)

# Basic workflow: Estado

```
$ svn status
M    global.h
A    docs
A    docs/README.txt
A    Makefile
```

```
$ svn status -v
      4      4 daniel      .
      4      4 daniel      main.c
M     4      4 daniel      global.h
A     0      ?   ?        docs
A     0      ?   ?        docs/README.txt
A     0      ?   ?        Makefile
```



# Basic workflow: Subir cambios

```
$ svn commit -m "Nuevos ficheros"
Adding          Makefile
Adding          docs
Adding          docs/README.txt
Sending         global.h
svn: Commit failed (details follow):
svn: Out of date: '/trunk/global.h' in transaction '5-1'
```

- Ha fallado !
- Porque global.h está desactualizado
- svn hace transacciones atómicas así que no ha hecho cambios en el repositorio
- Contactando con el repositorio:

```
$ svn status -u
      *          4   main.c
M     *          4   global.h
A     *          0   docs/README.txt
A     *          0   docs
A     *          0   Makefile
Status against revision:      5
```

- El repositorio está en la versión 5 y global.h y main.c están modificados frente a nuestra versión

# Basic workflow: Ver cambios

```
$ svn diff -r BASE:HEAD
```

```
Index: main.c
```

```
=====
```

```
--- main.c      (working copy)
```

```
+++ main.c      (revision 5)
```

```
@@ -6,5 +6,6 @@
```

```
{
```

```
    printf("%s\n\n",TITULO);
```

```
    printf("Hola mundo!\n");
```

```
+    printf("Y adios\n");
```

```
    exit(0);
```

```
}
```

```
Index: global.h
```

```
=====
```

```
--- global.h    (working copy)
```

```
+++ global.h    (revision 5)
```

```
@@ -1 +1 @@
```

```
+#define        TITULO "Mi Hola Mundo"
```

```
+#define        TITULO "Mi primer Hola Mundo"
```

- La opción -r indica a diff que nos compare dos versiones del repositorio
- BASE es la versión que tenemos, HEAD es la última en el repositorio (en este caso sería equivalente 4:5)

# Basic workflow: Ver comentarios

```
$ svn log -r BASE:HEAD
```

```
-----  
r4 | daniel | 2007-07-18 16:33:53 +0200 (Wed, 18 Jul 2007) | 2 lines
```

```
Importacion inicial
```

```
-----  
r5 | daniel | 2007-07-18 17:03:48 +0200 (Wed, 18 Jul 2007) | 1 line
```

```
Pongo una despedida y cambio el saludo  
-----
```

# Basic workflow: Actualizarse

```
$ svn update
U    main.c
C    global.h
Updated to revision 5.
```

- main.c se actualiza sin problemas pero global.h tiene un conflicto

```
$ ls -l
total 48
-rw-r--r--  1 daniel  daniel   46 Jul 18 16:40 Makefile
drwxr-xr-x  4 daniel  daniel  136 Jul 18 16:41 docs
-rw-r--r--  1 daniel  daniel  129 Jul 18 17:08 global.h
-rw-r--r--  1 daniel  daniel   57 Jul 18 17:08 global.h.mine
-rw-r--r--  1 daniel  daniel   31 Jul 18 17:08 global.h.r4
-rw-r--r--  1 daniel  daniel   38 Jul 18 17:08 global.h.r5
-rw-r--r--  1 daniel  daniel  155 Jul 18 17:08 main.c
```

- Ficheros:
  - .mine : el fichero que teníamos localmente modificado
  - .rBASE : la versión original que hemos modificado localmente (sale del .svn, la versión BASE)
  - .rHEAD : la última versión del repositorio

# Basic workflow: Resolver conflicto

```
$ cat global.h
<<<<<<< .mine
#define TITULO "Mi Hola Mundo"
#define DESPEDIDA "Adios"
=====
#define TITULO "Mi primer Hola Mundo"
>>>>>>> .r5
```

- Tiene marcada la zona con conflicto (en este caso el fichero es tan pequeño que es todo él)
- Editamos el fichero resolviendo el conflicto:

```
$ cat global.h
#define TITULO "Mi primer Hola Mundo"
#define DESPEDIDA "Adios"
```

- Indicamos que se ha resuelto el conflicto:

```
$ svn resolved global.h
Resolved conflicted state of 'global.h'
```

# Basic workflow: Resolver conflicto

```
$ ls
Makefile          docs              global.h          main.c
```

- Han desaparecido los ficheros auxiliares del conflicto
- Ya podemos hacer commit:

```
$ svn commit -m "Nuevos ficheros"
Adding          Makefile
Adding          docs
Adding          docs/README.txt
Sending         global.h
Transmitting file data ...
Committed revision 6.
```

# Tags and branches

# Tags and branches

- Tags:
  - Versiones a fijar (1.0, 2.0, etc)
  - Pensadas para no modificarse
- Branches:
  - Se continúa el desarrollo en ellas
  - Cambios que van a tardar en funcionar y queremos seguir sincronizándolas con el repositorio

```
$ svn copy file:///Users/daniel/myrepos/trunk  
file:///Users/daniel/myrepos/tags/proy-v1
```



# Merging changes between branches

## Del trunk a un branch

- Se crea el branch (estábamos en la revisión 6)  

```
svn copy file:///Users/daniel/myrepos/trunk  
file:///Users/daniel/myrepos/branches/dev-1
```
- Hago cambios en el branch (en main.c)
- Ha habido cambios en el trunk que quiero incorporar en mi branch (también en main.c):

```
$ svn log -r 5:HEAD file:///Users/daniel/Myrepos/trunk
```

```
-----  
r5 | daniel | 2007-07-18 17:03:48 +0200 (Wed, 18 Jul 2007) | 1 line
```

```
Pongo una despedida y cambio el saludo
```

```
-----  
r6 | daniel | 2007-07-18 17:17:05 +0200 (Wed, 18 Jul 2007) | 1 line
```

```
Nuevos ficheros
```

```
-----  
r9 | daniel | 2007-07-18 17:56:41 +0200 (Wed, 18 Jul 2007) | 1 line
```

```
Modifico la despedida
```

# Merging changes between branches

- Incorporar los cambios:

```
$ svn merge file:///Users/daniel/myrepos/trunk@6  
file:///Users/daniel/myrepos/trunk@9
```

```
U main.c
```

- Ó

```
$ svn merge -r 6:9 file:///Users/daniel/myrepos/trunk  
file:///Users/daniel/myrepos/trunk
```

- Le he indicado que incorpore al working directory (que es mi checkout del branch) las modificaciones que ha habido en el trunk desde la revisión 6 a la 9
- Faltaría hacer commit de los cambios al branch

# Merging changes between branches

## De un branch a otro (o al trunk)

- Se crea el branch
- Hago cambios en el branch
- Incorporar los cambios del branch en otro directorio

# Otras utilidades

# Copiar, mover, borrar

## Copiar:

### 1. A mano

```
$ cp global.h includes.h
```

```
$ svn add includes.h
```

### 2. A máquina

```
$ svn copy global.h includes.h
```

### 3. Remoto (file:/// o svn+ssh:///)

```
$ svn copy file:///Users/home/daniel/myrepos/trunk/global.h  
includes.h
```

## Mover (a máquina):

```
$ svn move include.c otronombre.c
```

## Borrar (a máquina):

```
$ svn delete otronombre.c
```

# Lock/Unlock

- Para ficheros que no se lleven bien con los merges (binarios)

```
$ svn lock fichero
```

```
$ svn unlock fichero
```

# Export

- Obtener un checkout sin los .svn

```
$ svn export URL
```

# Creación de clave privada

```
$ ssh-keygen -t dsa
```

- Creará un par de ficheros:
  - `id_dsa`
  - `id_dsa.pub`
- Pide una passphrase: dejar vacía si queremos hacer login sin que nos la pida
- El fichero `id_dsa` debe ir al directorio `.ssh` de la máquina origen
- El fichero `id_dsa.pub` debe ir al fichero `authorized_keys` del directorio `.ssh` de la máquina destino
- La primera vez nos pedirá aceptar el fingerprint de la máquina destino



# Referencias

- <http://subversion.tigris.org/>
- Libro online: <http://svnbook.red-bean.com/>
- D. Berlin y G. Rooney, “Practical Subversion”, Ed. Apress

# Version control with Subversion

Area de Ingeniería Telemática  
<http://www.tlm.unavarra.es>

Grupo de Redes, Sistemas y Servicios Telemáticos