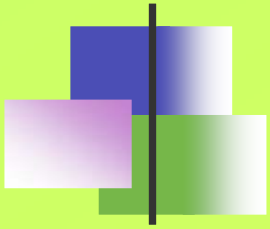
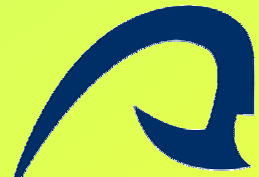


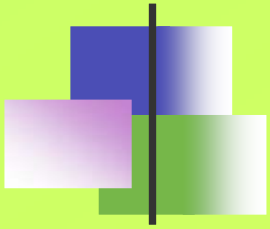
Sistemas de Control de Versiones

Subversion

Alexis Quesada Arencibia
Francisco J. Santana Pérez

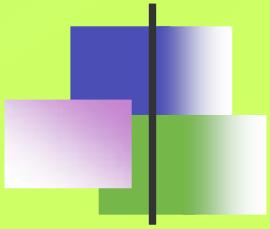


UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela Universitaria de Informática



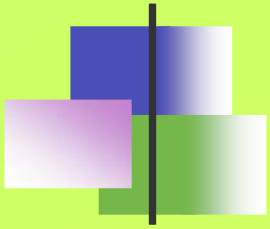
¿ Qué son ?

- Programas que permiten gestionar un repositorio de archivos y sus distintas versiones
- Utilizan una arquitectura cliente-servidor
  - Un servidor guarda la(s) versión(es) actual(es) del proyecto y su historia



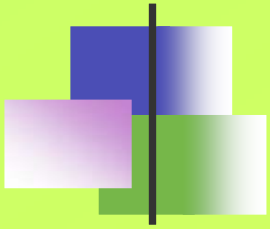
Nos permiten …

- Varios clientes pueden sacar copias del proyecto al mismo tiempo
- Realizar cambios a los ficheros manteniendo un histórico de los cambios
  - Deshacer los cambios hechos en un momento dado
  - Recuperar versiones pasadas
  - Ver históricos de cambios y comentarios
  - Los clientes pueden también comparar diferentes versiones de archivos



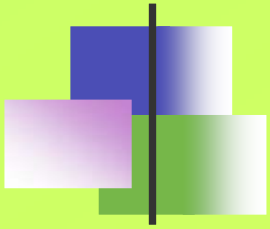
Nos permiten …

- Unir cambios realizados por diferentes usuarios sobre los mismos ficheros
- Sacar una "foto" histórica del proyecto tal como se encontraba en un momento determinado
- Actualizar una copia local con la última versión que se encuentra en el servidor
  - Esto elimina la necesidad de repetir las descargas del proyecto completo
- Mantener distintas ramas de un proyecto



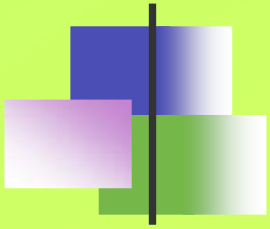
Sistemas de control de Versiones (SCV)

- **CVS**
  - **http://www.nongnu.org/cvs/**
- **Subversion**
  - **http://subversion.tigris.org/**
- **Arch**
  - **http://www.gnu.org/software/gnu-arch/**
- **DARCS**
  - **http://www.darcs.net/**
- **BitKeeper**
  - **http://www.bitkeeper.com/**



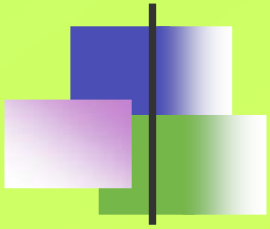
Subversion

- Sistema de control de versiones libre y de código abierto (2001)
- Está construido sobre una capa de portabilidad llamada *APR* (la biblioteca *Apache Portable Runtime*)
  - Debería funcionar sobre cualquier SO donde lo haga el servidor httpd Apache (Windows, Linux, Mac OS X, ...)



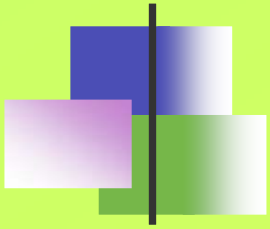
Subversion

- Es un sistema general que puede ser usado para administrar cualquier conjunto de ficheros
  - Texto, imágenes, video, ejecutables, ...
- El repositorio en el servidor consiste en una base de datos (*Berkeley DB*)



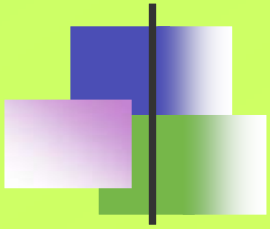
Características

- Versionado de directorios
- Facilidad para añadir/eliminar/mover ficheros y directorios
- Envíos atómicos
- Versionado de metadatos
- Soporte para binarios



Características

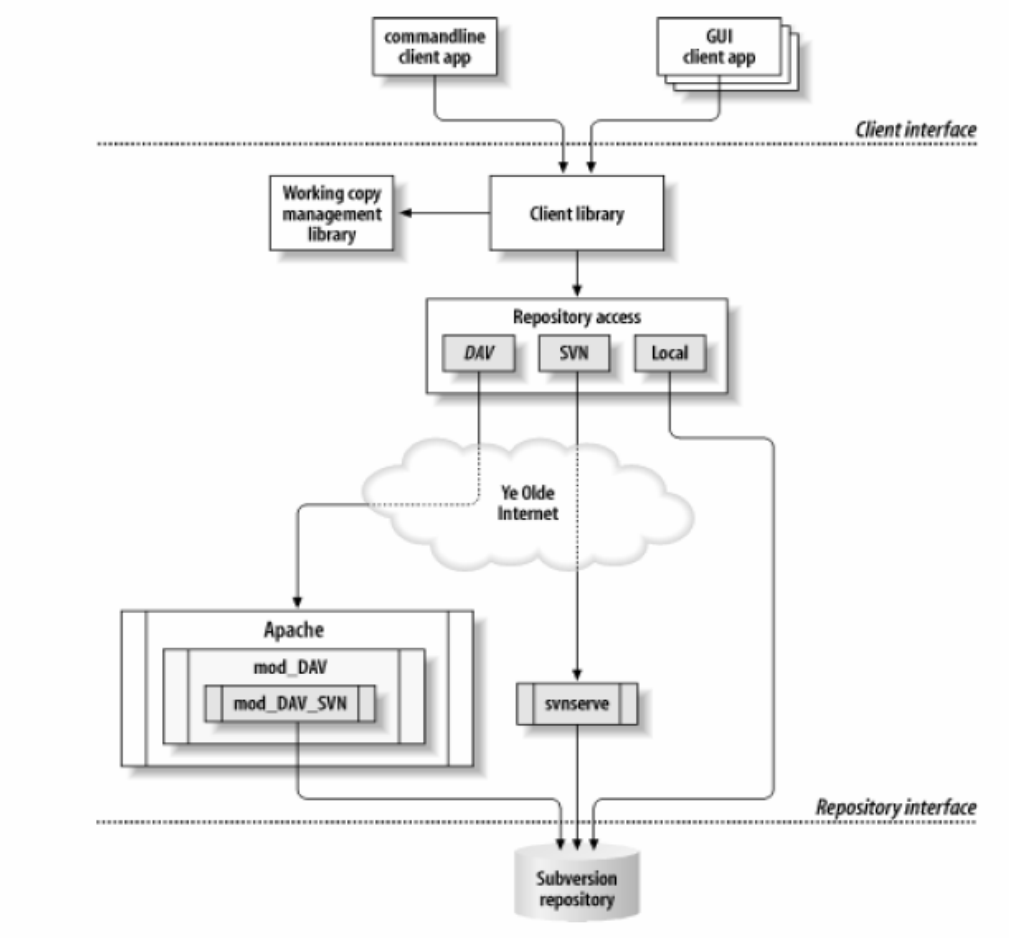
- Diferentes sistemas de acceso
  - **file://** acceso al sistema local
  - **http://** y **https://** haciendo uso del módulo DAV en Apache 2
  - **svn://** para svnserve y **svn+ssh://** para utilizar ssh
- Ramificación y etiquetado eficientes
- Se envían solo las diferencias en ambas direcciones

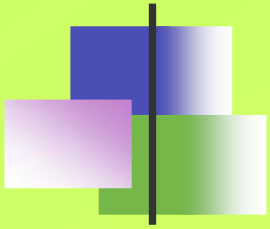


Características

- Existen clientes gráficos tanto para linux como para windows
  - Windows: *tortoiseSVN*
    - **http://tortoisesvn.tigris.org/**
  - Windows, Linux, Mac OS: *rapidsvn*
    - **http://rapidsvn.tigris.org/**

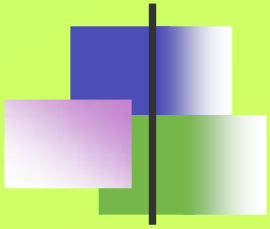
Arquitectura





Ciclo básico de trabajo

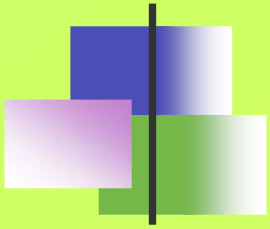
- **svn**: programa cliente de línea de órdenes
- La mejor ayuda:
  - **svn help *orden***



Ciclo básico de trabajo

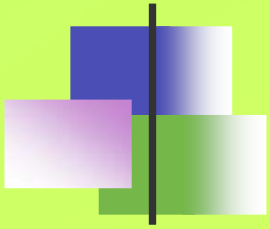
- Descargando un repositorio (crea una copia en la máquina local)
  - svn checkout URL_REPOSITORIO

```
$ svn checkout http://svn.collab.net/repos/svn/trunk subv
A subv/subversion.dsw
A subv/svn_check.dsp
…
Checked out revision 2499.
```



Ciclo básico de trabajo

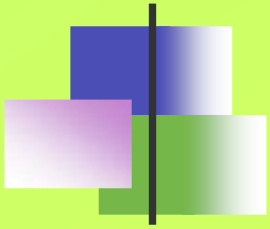
- Haciendo cambios en la copia de trabajo local
  - Cambios en los ficheros
    - No hay que indicarle nada a subversion. Detectará los cambios automáticamente
  - Cambios en el arbol
    - Añadir/eliminar/copiar/mover ficheros y directorios



Ciclo básico de trabajo

- Añadir/eliminar/copiar/mover ficheros y directorios
  - Los cambios son marcados en la copia local y serán reflejados en el repositorio cuando se envíen los cambios

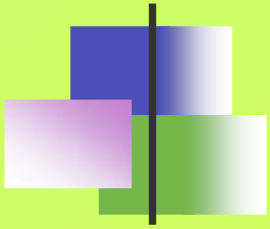
```
$ svn add fichOdir
$ svn delete fichOdir
$ svn copy fichOdir1 fichOdir2
$ svn move fichOdir1 fichOdir2
```



Ciclo básico de trabajo

- Examinando los cambios antes de enviarlos al repositorio
  - Se puede ejecutar en la raiz de la copia de trabajo para ver todo los cambios o en sobre una ruta o fichero esfecífica)

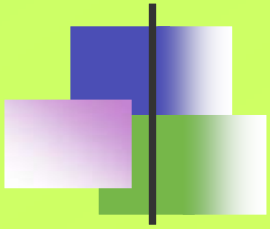
```
$ svn status
  L     abc.c                 # svn has a lock in its .svn directory for abc.c
M       bar.c                 # the content in bar.c has local modifications
 M      baz.c                 # baz.c has property but no content modifications
X       3rd_party             # this dir is part of an externals definition
?       foo.o                 # svn doesn't manage foo.o
!       some_dir              # svn manages this, but it's either missing or incomplet
~       qux                   # versioned as dir, but is file, or vice versa
I       .screenrc             # this file is ignored
A  +    moved_dir             # added with history of where it came from
M  +    moved_dir/README      # added with history and has local modifications
D       stuff/fish.c          # this file is scheduled for deletion
A       stuff/loot/bloo.h     # this file is scheduled for addition
C       stuff/loot/lump.c     # this file has conflicts from an update
     S  stuff/squawk          # this file or dir has been switched to a branch
…
```



Ciclo básico de trabajo

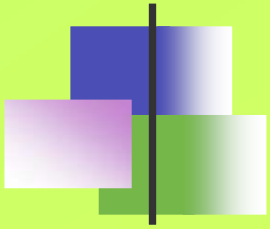
- Otra forma de examinar los cambios
  - svn diff

```
$ svn diff
Index: bar.c
===================================================================
--- bar.c (revision 3)
+++ bar.c (working copy)
@@ -1,7 +1,12 @@
+#include <sys/types.h>
+#include <sys/stat.h>
+#include <unistd.h>
+
+#include <stdio.h>

 int main(void) {
-  printf("Sixty-four slices of American Cheese...\n");
+  printf("Sixty-five slices of American Cheese...\n");
 return 0;
 }
```



Ciclo básico de trabajo

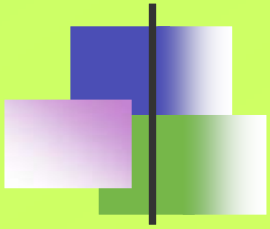
- Me arrepiento de mis ..... (deshacer las modificaciones locales)
  - svn revert PATH
- Para deshacer todo los cambios en un directorio se debe usar la opción
  - *--recursive*



Ciclo básico de trabajo

- Enviando los cambios al repositorio
  - svn commit [PATH] –m “mensaje de registro”
  - svn commit [PATH] –F fichero_log

```
$ svn commit --message "Corrected number of cheese slices."
Sending sandwich.txt
Transmitting file data .
Committed revision 3.
```

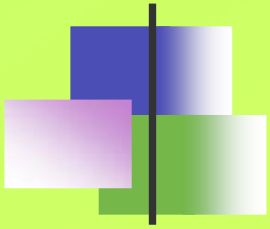


Ciclo básico de trabajo

- Actualizar la copia de trabajo local
  - svn update

```
$ svn update
U foo.c
U bar.c
Updated to revision 2.
```

U => Actualizado
A => Añadido
D => Eliminado
R => Reemplazado
G => Fundido
C => Conflicto



Ciclo básico de trabajo

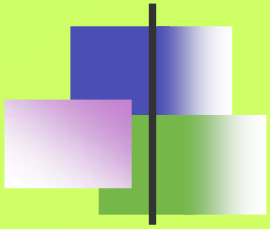
- Resolviendo conflictos

```
$ svn update
U INSTALL
G README
C bar.c
Updated to revision 46.
```

Actualizado !!! No hay problema

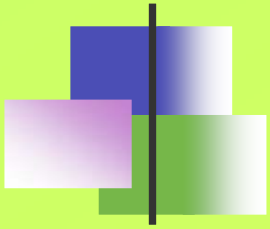
Fundido !!! Nos entendemos: ¡ tu a tus cosas y yo a las mías !

Aquí llegó la bronca !!!!



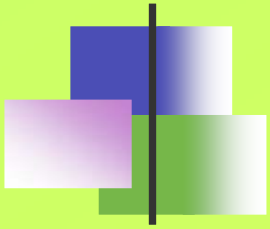
Ciclo básico de trabajo

- Pero dicen que hablando se entiende la gente !!!
- Acto de conciliación !!!
  - Ayuda de subversion en relación a los conflictos
    - Nos informa (C)
    - Coloca marcas de conflicto en el fichero para demostrar visualmente las áreas solapadas
    - Para cada fichero en conflicto, subversion coloca tres ficheros temporales extra en la copia de trabajo local
      - *Filename.mine, filename.rOLDREV, filename.RNEWREV*



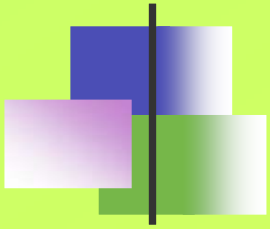
Ciclo básico de trabajo

- Soluciones:
  - Fusionar el texto en conflicto a mano
  - Copiar uno de los ficheros temporales sobre su fichero de trabajo
  - Ejecutar svn revert para eliminar los cambios locales



Ciclo básico de trabajo

- Subversion no permitirá enviar el fichero en conflicto hasta que los tres ficheros temporales sean borrados
- Una vez solucionado el conflicto, es necesario comunicarselo a subversion
  - svn resolved PATH
  - Esto borrará los tres ficheros temporales y Subversion no considerará por mas tiempo que el fichero está en estado de conflicto

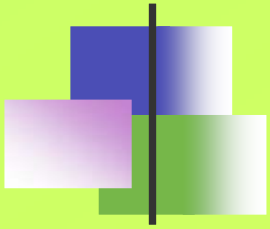


Ciclo básico de trabajo

- Fusionar el texto en conflicto a mano

```
$ cat sandwich.txt
Top piece of bread
Mayonnaise
Lettuce
Tomato
Provolone
<<<<<<< .mine
Salami
Mortadella
Prosciutto
=======
Sauerkraut
Grilled Chicken
>>>>>>> .r2
Creole Mustard
Bottom piece of bread
```

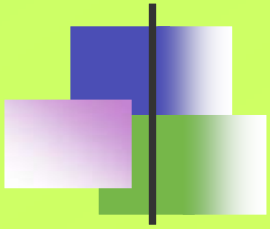
- svn resolved fichero



Ciclo básico de trabajo

- Copiar uno de los ficheros temporales sobre su fichero de trabajo

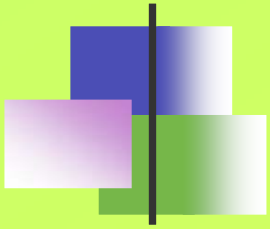
```
$ svn update
C sandwich.txt
Updated to revision 2.
$ ls sandwich.*
sandwich.txt sandwich.txt.mine sandwich.txt.r2
sandwich.txt.r1
$ cp sandwich.txt.r2 sandwich.txt
$ svn resolved sandwich.txt
```



Ciclo básico de trabajo

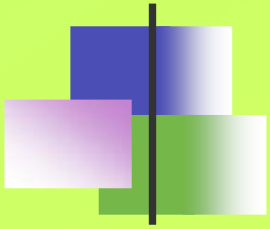
- Eliminar los cambios locales
  - svn revert fichero
  - Si se invierte un fichero en conflicto, no es necesario ejecutar “svn resolved”

```
$ svn revert sandwich.txt
Reverted 'sandwich.txt'
$ ls sandwich.*
sandwich.txt
```



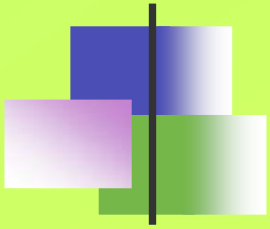
Ciclo básico de trabajo

- Una vez resueltos los conflictos, enviamos los cambios
  - svn commit –m “mensaje”



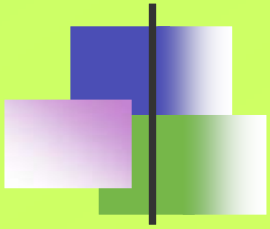
Ciclo básico de trabajo

- Examinando el historial
  - Mostrar información sobre los registros
    - **svn log**
  - Mostrar información sobre cambios
    - **svn diff**
  - Mostrar revisiones anteriores de ficheros
    - **svn cat**
  - Mostrar los ficheros de un directorio para una revisión concreta
    - **svn list**



Otras órdenes útiles

- Completar transacciones pendientes eliminando el estado de bloqueo
  - svn cleanup
- Copiar un árbol de ficheros sin versionar en el repositorio, creando directorios intermedios
  - svn import [PATH] URL
- Crear directorios en la copia local o repositorio
  - svn mkdir PATH_O_URL



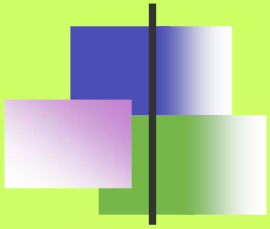
Creación de ramas (*branches*)

- svn cp ORIGEN DESTINO

```
svn cp file:///home/usuario/svn/holamundo/head
file:///home/usuario/svn/holamundo/ramas/holamundo-beta
```

- Y si corregimos errores en la rama, podemos rescatarlas en la versión de desarrollo
  - svn merge

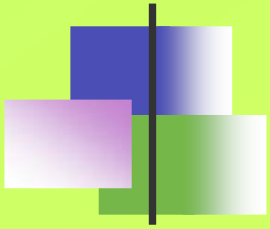
```
svn merge -r 176:177
file:///home/usuario/svn/holamundo/ramas/holamundo-beta
```



Etiquetas (tags)

- svn cp ORIGEN DESTINO

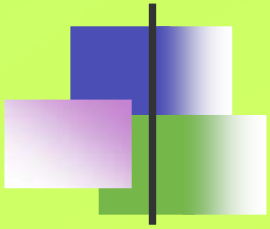
```
$ svn cp -r 345
file:///home/usuario/svn/holamundo/ramas/holamundo-beta \
file:///home/usuario/svn/holamundo/finales/1_0

$svn propset svn:read-only 1
file:///home/usuario/svn/holamundo/finales/1_0
```



Creando un repositorio

- svnadmin create REPO_PATH

```
[aquesada@dumby svn]$ svnadmin create ./aquesada
[aquesada@dumby svn]$ cd aquesada/
[aquesada@dumby aquesada]$ ls
conf  dav  db  format  hooks  locks  README.txt
[aquesada@dumby aquesada]$
```



Referencias

- http://subversion.tigris.org/
- http://svnbook.red-bean.com/
- http://es.wikipedia.org/wiki/Sistema_de_control_de_versi%C3%B3n
- http://acm2.asoc.fi.upm.es/~chernando/doc/svn/
- http://www.kikov.org/subversion-tutorial-es-index
- http://josecely.tecsua.com/?p=12
- *http://www.escomposlinux.org/fer_y_juanjo/index.php?pag=subversion.htm/*
- http://kopernix.com/?q=svnd_como