

Sistemas de Control de Fuentes



Website



Índice

Introducción

Funciones

Why Use It ?

Problema

Elementos

Instalación

Operaciones

Ejemplos

Introducción

- Casos Verídicos
 - Carlos, ¿Podrías pasarme una copia con los cambios que hiciste ayer ?
 - Ana, esta función da error, ¿Has cambiado algo?
 - He perdido la clase Empleado que implemente el lunes en practicas... Otra vez a implementar...
 - Mañana quedamos en la biblioteca y terminamos la practica entre los dos.

Introducción

- ¿A que se debe este tipo de situaciones?
 - Proyecto/Prácticas en común **desorganizadas**
 - Proceso donde la información **no esta bajo control.**



Introducción

- Proyecto => Complejo
- Proyecto => “Varios” Desarrolladores
- Proyecto => Control de Información
- Proyecto => Herramientas / Técnicas
- Proyecto => Mantener Código Fuente

Es fundamental un **sistema de control de fuentes**

Funciones

- Hacer un **seguimiento** del estado de cada archivos a través del tiempo.
- **Combinar** contribuciones de varios desarrolladores.
- Almacenar un **historial de cambios** realizados a través del tiempo por diferentes personas.

Funciones

- “El control de versiones es el arte del manejo de los cambios en la información”
- **Retroceder** y ver los cambios de los archivos antes de que se aplicaran
- Facilita el **trabajo colaborativo**

Funciones

- El control de versiones **no sólo** es necesario para el software



Why Use It ?

- Integridad de los Datos
 - Historial de Cambios
 - Repositorio Central
 - Favorece las copias de seguridad
 - Posibilidad de fallo en un disco local

Why Use It ?

- Productividad
 - No se pierde tiempo en la integración del trabajo de varios desarrolladores.
 - Facilita el acceso a los datos de los desarrolladores.

Why Use It ?

- **Control** y **Distribución** de Trabajo
 - ¿**Quién** lo hace?
 - ¿**Qué** hace?
 - ¿**Cuándo** lo hace?

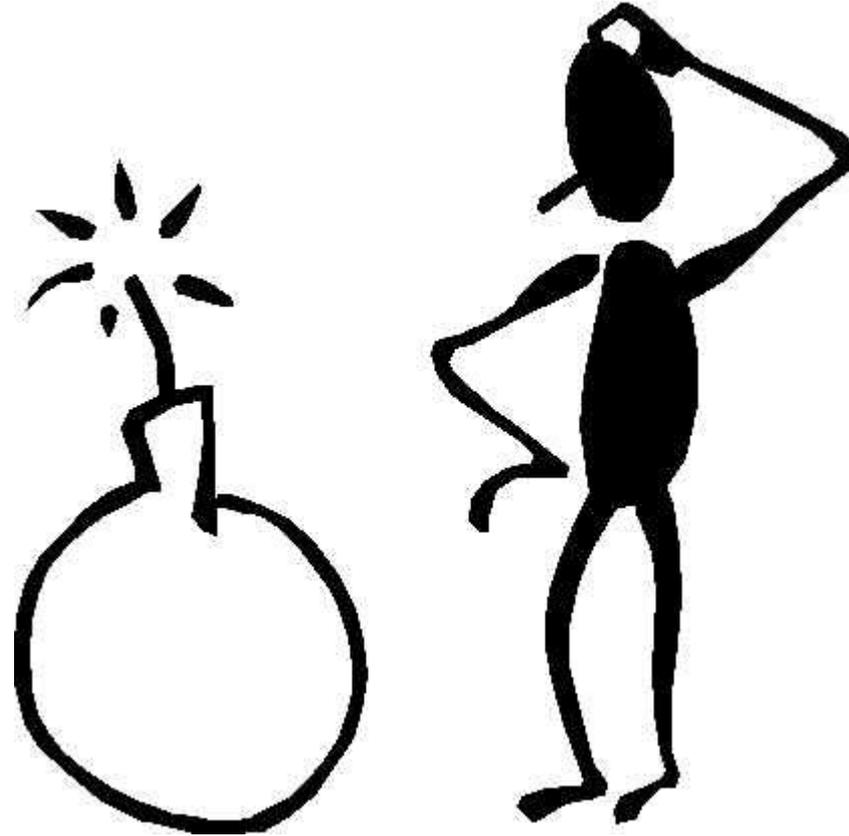
Why Use It ?

- Lineas de Desarrollo (Ramas)
 - Facilita el mantenimiento de varias lineas
 - Ayuda a combinar diferentes lineas
 - Al añadir nuevas características a una linea, estas se pueden propagar fácilmente a otras lineas de desarrollo.

Why Use It ?

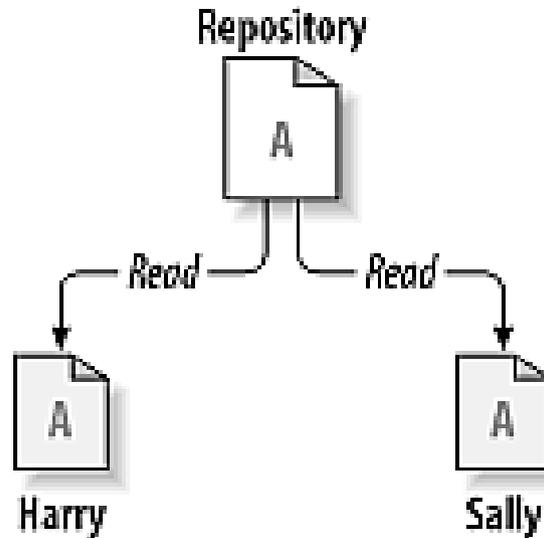
- **Herramienta** para la Ingeniería del Software.
 - Un proceso de desarrollo de software necesita de ciertas herramientas para lograr su objetivo.
 - **Subversion** es un proyecto de código abierto alojado por la comunidad *Tigris.org* que desarrolla diversas herramientas enfocadas a la ingeniería del software.

El Problema

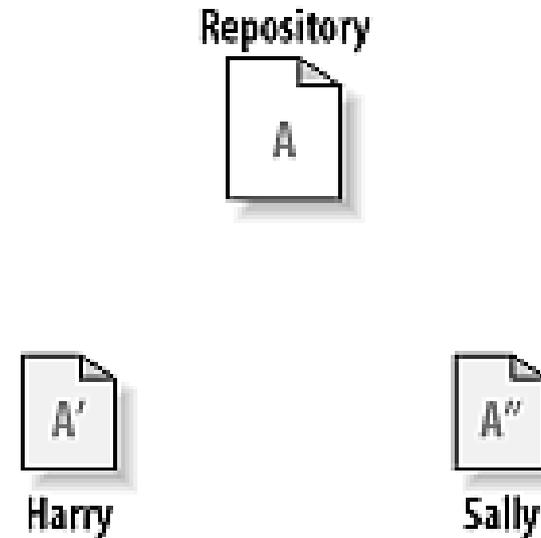


Sobreescribir

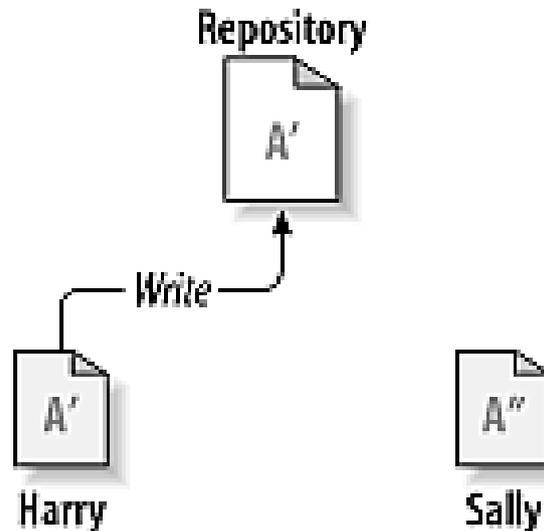
Two users read the same file



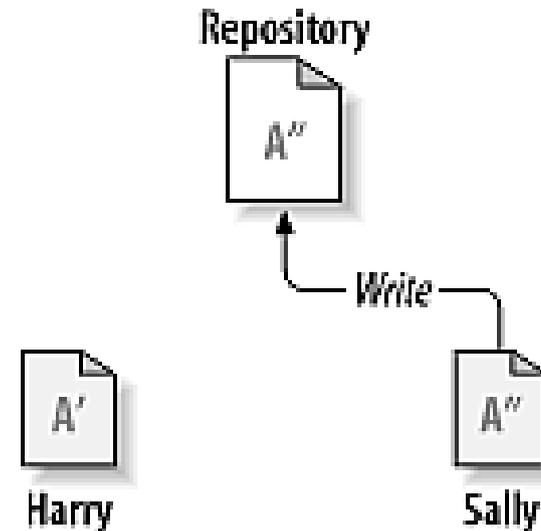
They both begin to edit their copies



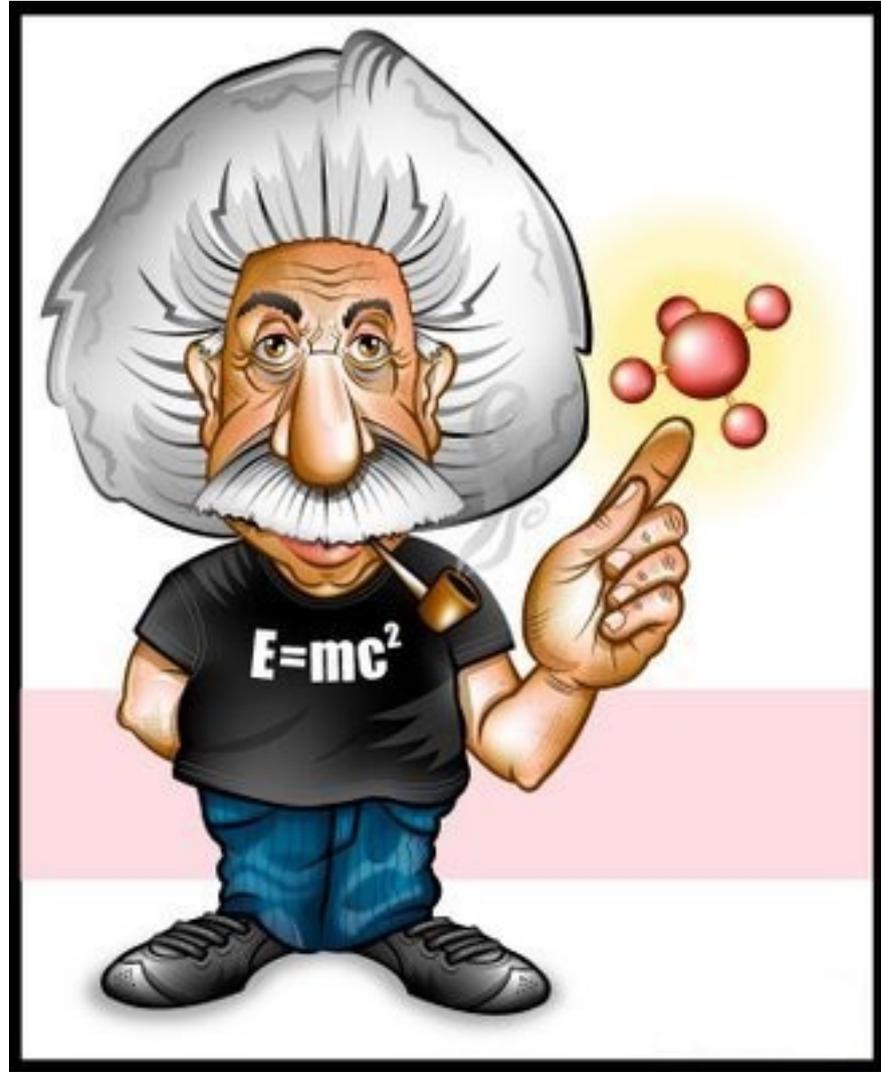
Harry publishes his version first



Sally accidentally overwrites Harry's version

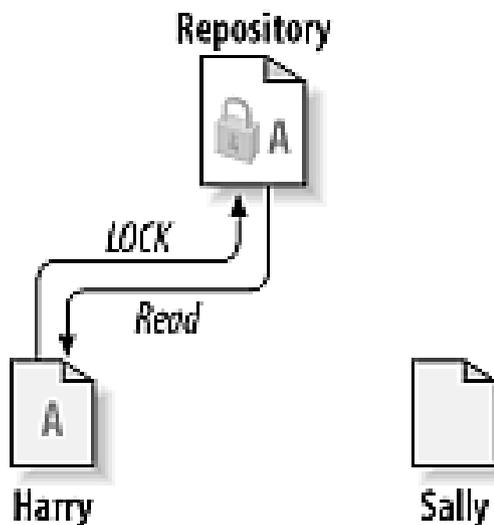


La Solución

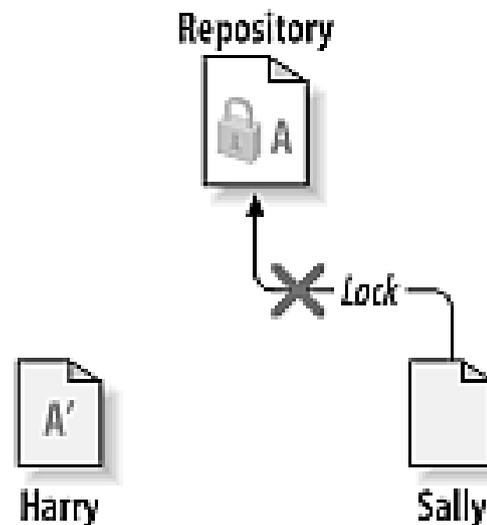


Lock-Modify-Unlock

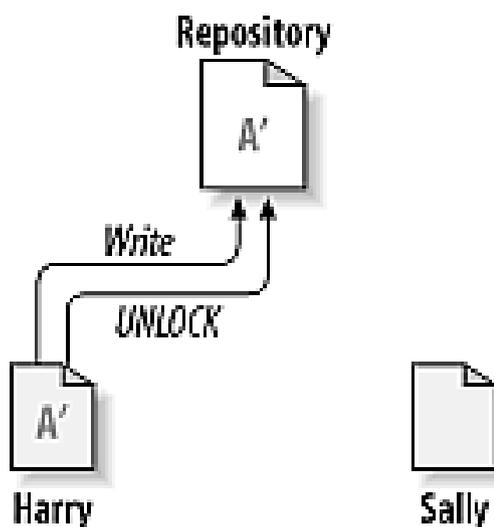
Harry "locks" file A, then copies it for editing



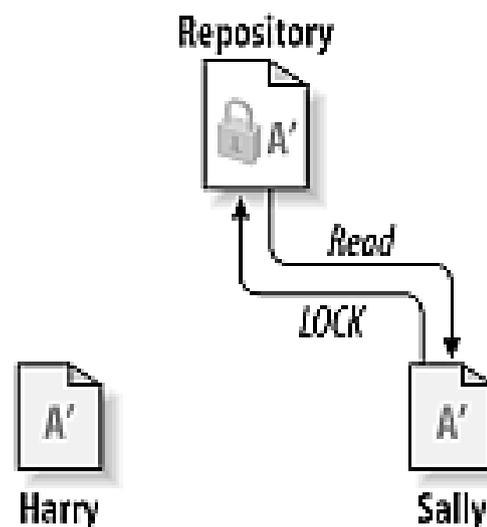
While Harry edits, Sally's lock attempt fails



Harry writes his version, then releases his lock

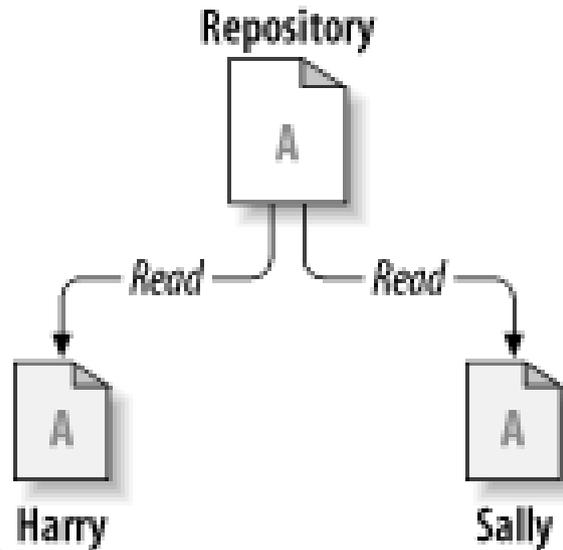


Now Sally can lock, read, and edit the latest version

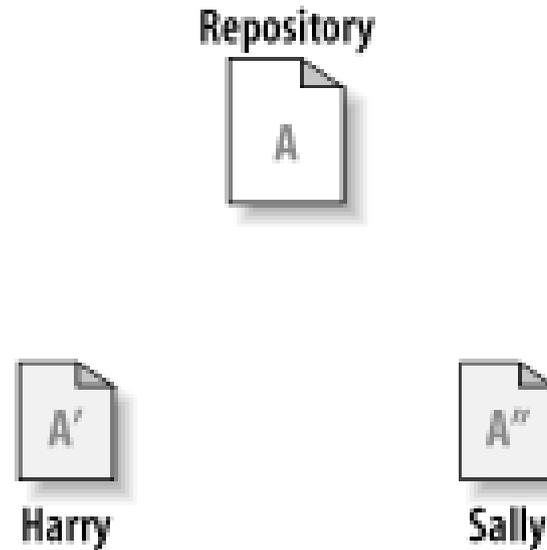


Copy-Modify-Merge

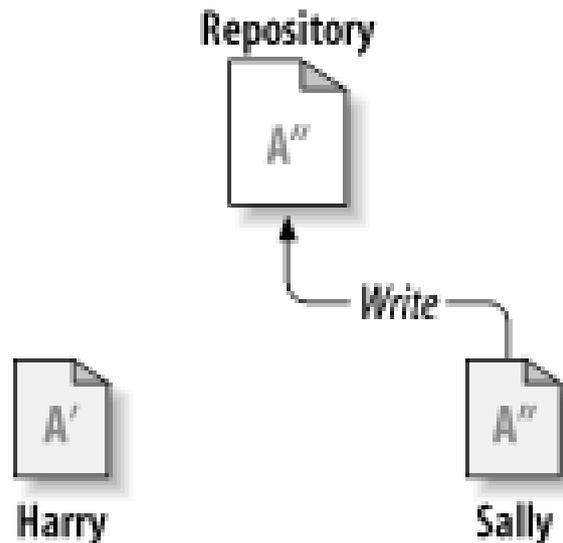
Two users copy the same file



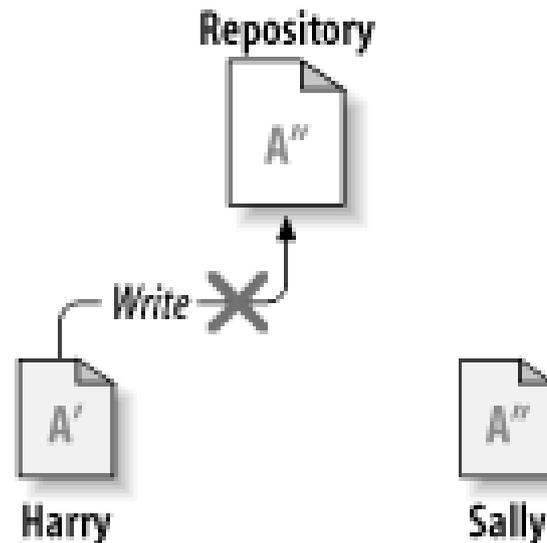
They both begin to edit their copies



Sally publishes her version first



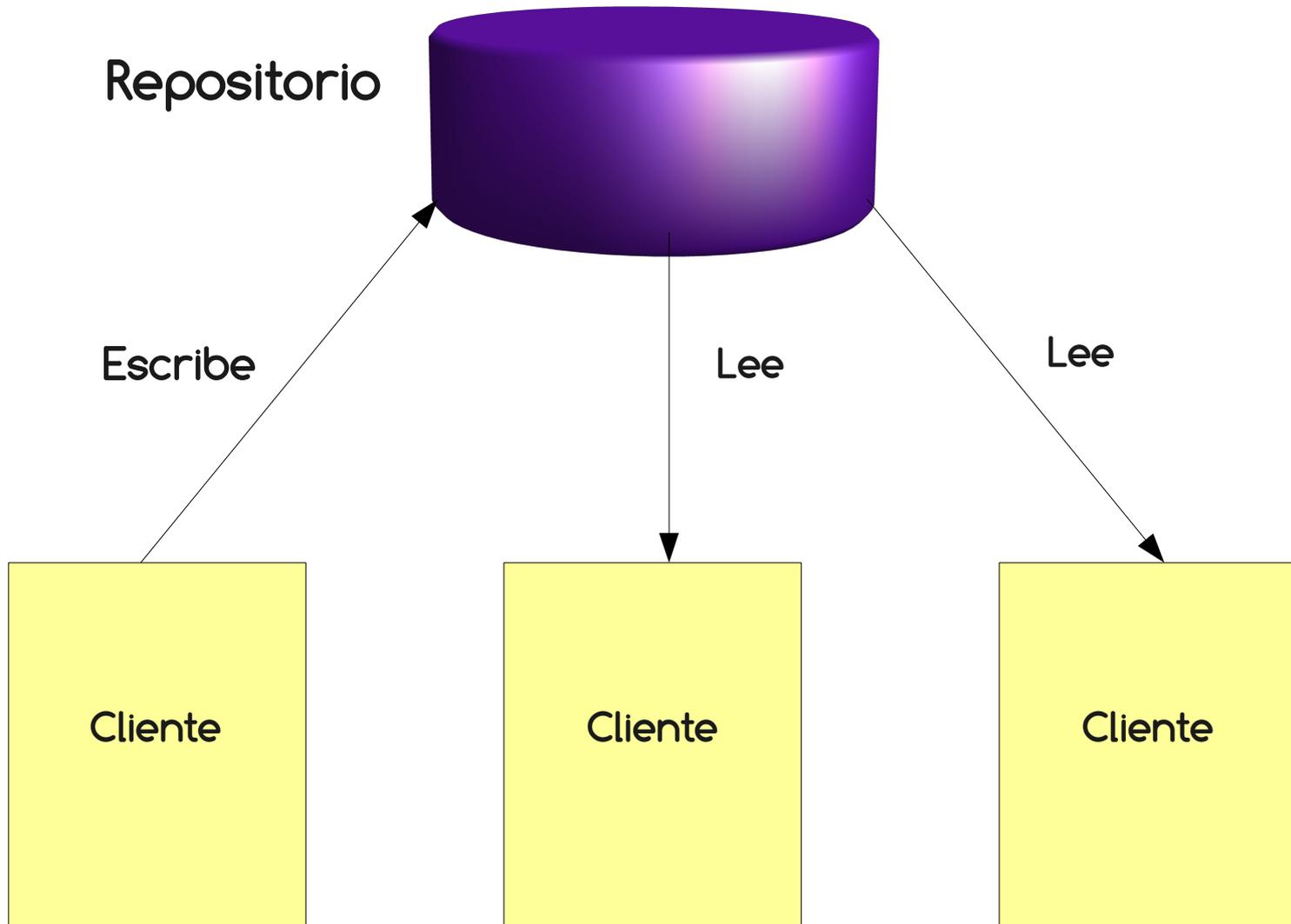
Harry gets an "out-of-date" error



Elementos

- Un **repositorio**, depósito o archivo es un lugar centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos
- Subversion es un repositorio en forma de árbol con una jerarquía de directorios y archivos.

Elementos



Elementos

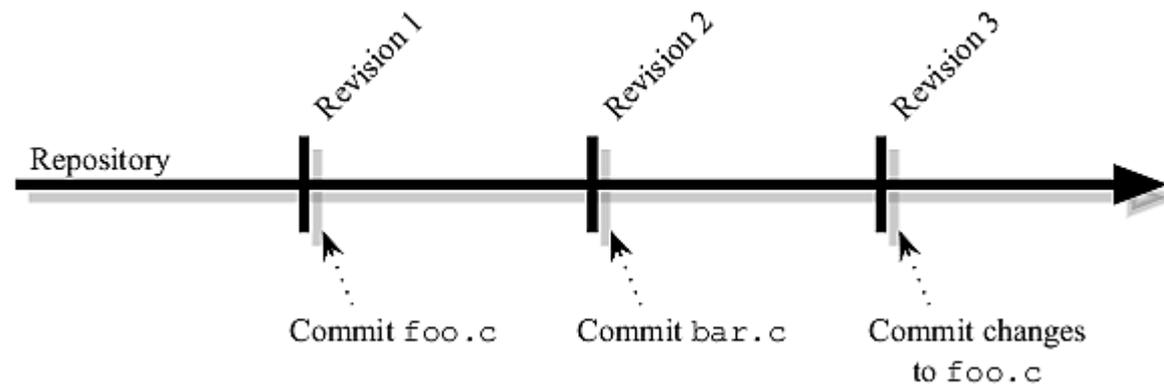
- Repositorio **Centralizado**
 - Mas sencillo de usar que los distribuidos
 - Control total sobre las versiones
 - Limitaciones en el acceso
 - Se reduce el número de conflictos

Elementos

- **Copia de Trabajo**
 - Copia Local de los archivos y directorios del repositorio. A ella se incorporarán los cambios o pondrán a disposición los tuyos cuando tú se lo indiques.
 - Directorio `./svn`

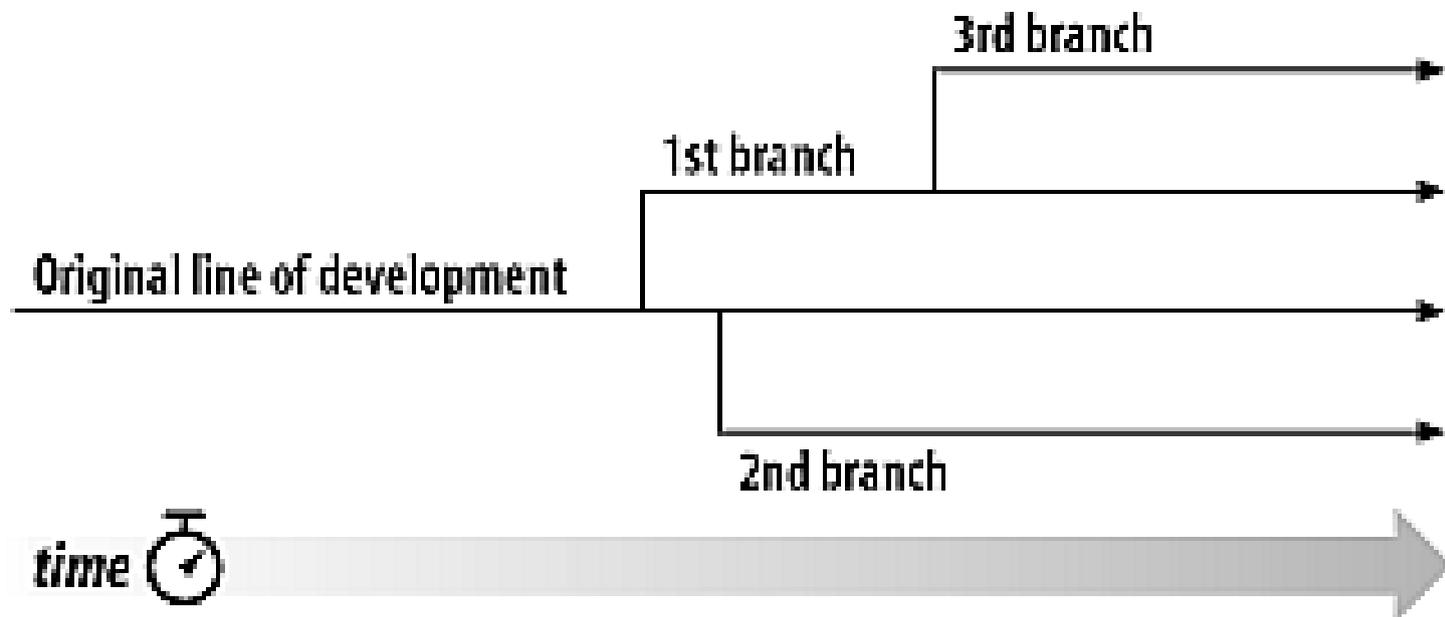
Elementos

- **Revisión**
 - Es una “instantánea” del repositorio en un momento particular en el tiempo, cada vez que el servidor acepta un envío se crea una nueva revisión.
 - Para identificarlas se utilizan números comenzando desde cero.



Elementos

- **Ramas (Branch)**
 - Lineas de desarrollo independientes a otras, pero comparten una historia común si mira lo suficientemente atrás en el tiempo.



Elementos

- Etiquetas (Tags)
 - Es una rama sobre la que **no se incluyen** cambios
 - Contiene el código/ficheros de una **revisión específica.**
 - Etiquetamos la revisión para establecer un hito en la línea de desarrollo.

Instalación

- Debes elegir tu camino....
- La pastilla **roja**
 - `sudo apt-get install subversion`
- La pastilla **azul**
 - RapidSVN
 - TortoiseSVN

<http://subversion.apache.org/packages.html>

Instalación

- Netbeans
 - Plug-in subversion
 - [Guia de Uso](#)
- Eclipse
 - Subclipse
 - [Website](#)

Instalación



General



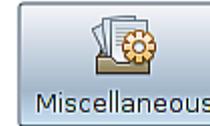
Editor



Fonts & Colors



Keymap



Miscellaneous

Ant Appearance Diff Files GUI Builder Java Debugger Tasks **Versioning**

Versioning Systems:

- CVS
- Local History
- Mercurial
- Subversion**

Path to the SVN executable File:

(e.g., /usr/local/bin)

Connection Settings

Use this Button to manage the Connection Settings for all previously used URLs:

Subversion Status Labels

Format:

To define the Folder variable content (Branch/Tag name) press:

Other

Automatically open Output window tabs

Operaciones

- **svn --version**
- **svn help** *subcomando*
 - *add*
 - *import*
 - *checkout*

Operaciones

- **svn checkout**
<https://svnosl.googlecode.com/svn/svnosl> --username mi_usuario

```
A   svnosl/trunk
A   svnosl/trunk/plantilla.html
A   svnosl/branches
A   svnosl/branches/rama01
A   svnosl/branches/rama01/hola.cpp
A   svnosl/branches/rama01/leeme.txt
A   svnosl/branches/rama01/hola.h
A   svnosl/branches/rama01/img
A   svnosl/tags
A   svnosl/tags/miproyecto1.0
A   svnosl/tags/miproyecto1.0/hola.cpp
A   svnosl/tags/miproyecto1.0/leeme.txt
A   svnosl/tags/miproyecto1.0/hola.h
A   svnosl/tags/miproyecto1.0/img
Revisión obtenida: 24
```

Operaciones

- **checkout** : Obtener una copia local de la última revisión
- --username **Solo en el checkout inicial**
- Aparece un directorio **svnosl** que contiene los ficheros del repositorio
- Se lanzan las ordenes sobre ese directorio

Operaciones

- Estructura común del repositorio
 - **/trunk** Contiene la línea principal desarrollo
 - **/branch** Contiene las ramas
 - **/tags** Contiene los tags
- **svn checkout**
`https://svnosl.googlecode.com/svn/trunk`
`svnosl --username mi_usuario`

Ciclo de Trabajo

- **Actualizar** la copia de trabajo
- **Modificar** la copia de trabajo
- **Comprobar** los cambios realizados
- **Deshacer** ciertos cambios
- **Resolver** conflictos
- **Guardar** los cambios realizados

Actualizar la copia de Trabajo

- `svn update`
- Indica los cambios que existen entre nuestra copia de trabajo y el repositorio, **con respecto a la ultima revisión**
- `svn update --revision 23`

```
A  Added
D  Deleted
U  Updated
C  Conflict
G  Merged
E  Existed
```

Actualizar la copia de Trabajo

```
murdock@A-Team:~/svnosl$ svn update --revision 23
A   asistencia.txt
A   asignaturas
A   hola.cpp
A   leeme.txt
A   fichero_nano.txt
A   hola.h
A   img
Actualizado a la revisión 23.
murdock@A-Team:~/svnosl$ svn update
D   asistencia.txt
D   asignaturas
D   hola.cpp
D   leeme.txt
D   fichero_nano.txt
D   hola.h
D   img
Actualizado a la revisión 24.
```

Modificar la copia de Trabajo

- `svn add fichero`
- `svn delete fichero`
- `svn copy fichero`
- `svn move fichero`
- `svn mkdir directorio`
 - Crea el directorio y lo añade (`mkdir + add`)

```
murdock@A-Team:~/svnosl$ nano
murdock@A-Team:~/svnosl$ svn add ejemplo.cpp
A      ejemplo.cpp
murdock@A-Team:~/svnosl$ nano
murdock@A-Team:~/svnosl$ svn add ejemplo.h
A      ejemplo.h
murdock@A-Team:~/svnosl$ █
```

Comprobar Cambios

- **svn status**
 - Muestras los ficheros donde hay cambios
- **svn status -q**
- **svn status -u**
- **svn status -v**

```
?    scratch.c          # file is not under version control
A    stuff/loot/bloo.h # file is scheduled for addition
C    stuff/loot/lump.c # file has textual conflicts from an update
D    stuff/fish.c     # file is scheduled for deletion
M    bar.c            # the content in bar.c has local modifications
```

Comprobar Cambios

```
murdock@A-Team:~/svnosl$ svn status
?      leeme.txt
?      fichero.txt
A      asistentes
murdock@A-Team:~/svnosl$ svn status -q
A      asistentes
murdock@A-Team:~/svnosl$ svn status -u
A      0      asistentes
?      leeme.txt
?      fichero.txt
Estado respecto a la revisión:    26
murdock@A-Team:~/svnosl$ svn status -v
      26      26 Fran.Lucena .
?      leeme.txt
?      fichero.txt
      26      26 Fran.Lucena ejemplo.h
A      0      ? ?      asistentes
      26      10 Fran.Lucena plantilla.html
      26      26 Fran.Lucena ejemplo.cpp
      26      25 Fran.Lucena SCS_CAs.pem
murdock@A-Team:~/svnosl$
```

Comprobar Cambios

- **svn diff**
 - Ver las diferencias con exactitud
 - Por defecto el ultimo fichero modificado
- **svn diff fichero**

Comprobar Cambios

```
murdock@A-Team:~/svnosl$ svn diff ejemplo.h
Index: ejemplo.h
=====
--- ejemplo.h      (revisión: 26)
+++ ejemplo.h      (copia de trabajo)
@@ -1 +1,2 @@
-lo que sea
+
+He eliminado la linea anterior.
murdock@A-Team:~/svnosl$
```

Comprobar Cambios

- Para una determinada revisión
- **svn diff -r 26 ejemplo.h**

```
murdock@A-Team:~/svnosl$ svn diff -r26 ejemplo.h
```

```
Index: ejemplo.h
```

```
=====
```

```
--- ejemplo.h      (revisión: 26)  
+++ ejemplo.h      (copia de trabajo)  
@@ -1 +1,2 @@  
-lo que sea  
+  
+He eliminado la linea anterior.
```

Deshacer Cambios

- **svn revert**
 - Mediante la orden diff se observa un error y se corrige el cambio mediante esta orden
 - Ejecutamos svn delete por error

Resolver Conflictos

- Aparecen cuando actualizamos la copia de trabajo

```
$ svn update
U  INSTALL
G  README
Conflict discovered in 'bar.c'.
Select: (p) postpone, (df) diff-full, (e) edit,
        (h) help for more options:
```

Resolver Conflictos

- Aparecen los siguientes ficheros
 - filename.mine Mi Archivo
 - filename.rOLDREV Version original
 - filename.rNEWREV Version repositorio

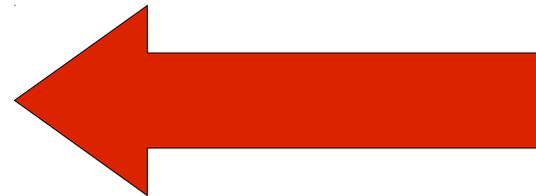
Resolver Conflictos

- Postpone (**svn no admite commit**)

```
$ svn update
Conflict discovered in 'sandwich.txt'.
Select: (p) postpone, (df) diff-full, (e) edit,
        (h)elp for more options : p
C sandwich.txt
Updated to revision 2.
$ ls -l
sandwich.txt
sandwich.txt.mine
sandwich.txt.r1
sandwich.txt.r2
```

Resolver Conflicts

```
..
Select: (p) postpone, (df) diff-full, (e) edit,
        (h)elp for more options : df
--- .svn/text-base/sandwich.txt.svn-base      Tue Dec 11 21:33:57 2007
+++ .svn/tmp/tempfile.32.tmp                  Tue Dec 11 21:34:33 2007
@@ -1 +1,5 @@
-Just buy a sandwich.
+<<<<<<< .mine
+Go pick up a cheesesteak.
+=====
+Bring me a taco!
+>>>>>>> .r32
..
```



Resolver Conflict

- By Hand

```
$ 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
```

Resolver Conflicts

- `svn resolve`

```
$ svn resolve --accept working sandwich.txt
```

```
Resolved conflicted state of 'sandwich.txt'
```

```
$ svn commit -m "Go ahead and use my sandwich, discarding Sally's edits."
```

Guardar Cambios

- `svn commit -m "Corregido makefile"`

```
murdock@A-Team:~/svnosl$ svn commit -m "Mas ficheros necesarios"
Añadiendo fichero.txt
Añadiendo leeme.txt
Transmitiendo contenido de archivos ..
Commit de la revisión 28.
```

Ciclo de Trabajo

- **Actualizar** la copia de trabajo
- **Modificar** la copia de trabajo
- **Comprobar** los cambios realizados
- **Deshacer** ciertos cambios
- **Resolver** conflictos
- **Guardar** los cambios realizados

Operaciones

- `svn log`
- `svn log -r 20 -v`
- `svn log -r 10:20`
- `svn log -r 20:15`

```
murdock@A-Team:~/svnosl$ svn log -r 20 -v
```

```
-----  
r20 | esloguin | 2009-12-17 21:13:56 +0100 (jue 17 de dic de 2009) | 3 lines
```

```
Rutas cambiadas:
```

```
  M /trunk/asistencia.txt
```

```
"Version antoñico"
```

Operaciones

- **svn list -v directorio (URL)**
 - Ficheros disponibles en el directorio

```
svn list -v http://svn.collab.net/repos/svn
20620  harry          1084 Jul 13  2006 README
23339  harry              Feb 04 01:40 branches/
21282  sally             Aug 27 09:41 developer-resources/
23198  harry             Jan 23 17:17 tags/
23351  sally             Feb 05 13:26 trunk/
```

Operaciones

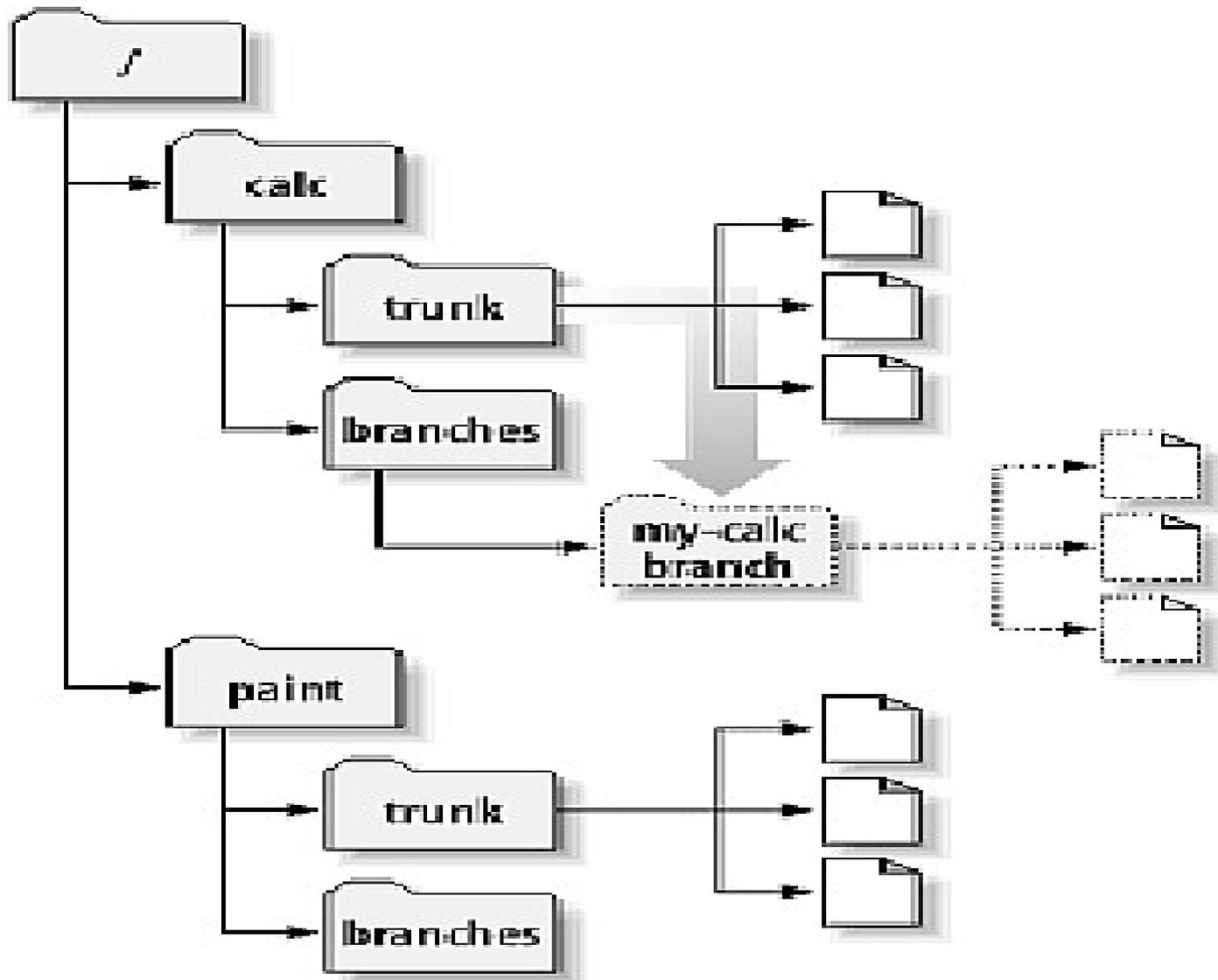
- Ramas
 - svn copy
 - svn merge
 - svn diff
 - svn switch

Operaciones

```
$ svn copy http://svn.example.com/repos/calc/trunk \  
           http://svn.example.com/repos/calc/branches/my-calc-branch \  
           -m "Creating a private branch of /calc/trunk."
```

```
Committed revision 341.
```

Operaciones



Operaciones

- Tags (**svn copy**)

```
$ svn copy http://svn.example.com/repos/calc/trunk \  
           http://svn.example.com/repos/calc/tags/release-1.0 \  
           -m "Tagging the 1.0 release of the 'calc' project."
```

```
Committed revision 902.
```

Operaciones

```
A   svnosl/trunk
A   svnosl/trunk/plantilla.html
A   svnosl/branches
A   svnosl/branches/rama01
A   svnosl/branches/rama01/hola.cpp
A   svnosl/branches/rama01/leeme.txt
A   svnosl/branches/rama01/hola.h
A   svnosl/branches/rama01/img
A   svnosl/tags
A   svnosl/tags/miproyecto1.0
A   svnosl/tags/miproyecto1.0/hola.cpp
A   svnosl/tags/miproyecto1.0/leeme.txt
A   svnosl/tags/miproyecto1.0/hola.h
A   svnosl/tags/miproyecto1.0/img
Revisión obtenida: 24
```

SUBVERSION



Licencia

Fco Javier Lucena Lucena

fran.lucena@gmail.com

<http://www.franlucena.es>

