

SVN

Víctor Herrero Cazurro

Contenido

1. ¿Qué es un sistema de control de versiones?
2. ¿Que es SVN?
3. Visual SVN
4. Tortoise SVN
5. Subversive
6. Modo de trabajo
7. Comandos
 1. Checkout
 2. Status
 3. Add
 4. Revert
 5. Commit
 6. Log
 7. Update
 8. Branch
 9. Merge
 10. Tag

¿Qué es un sistema de control de versiones?

- Herramienta que almacena todos los cambios que se hacen tanto en la estructura de directorios como en el contenido de los ficheros, con el fin de tener un histórico navegable.
- Estas herramientas son imprescindibles cuando más de una persona trabaja con los mismos archivos, para tener una trazabilidad del cambio, quién, cuándo, qué,...

¿Qué es un sistema de control de versiones?

- Estos sistemas deben ser capaces de comprobar el estado de los ficheros con el objetivo de que no se pierdan cambios realizados por un usuario, por ser pisados por cambios de otro usuario.

¿Qué es SVN?

- Herramienta de código abierto, multiplataforma, para el control de versiones de ficheros de texto plano.
- Se basa en un repositorio central que sirve los ficheros, recordando todos los cambios, el histórico, de todos los directorios y ficheros.
- Cada **Commit** (subida de cambios), provoca una nueva versión.

¿Qué es SVN?

- Es posible copiar y renombrar ficheros.
- Se pueden crear ramas, esto será un nuevo camino que se bifurca en el histórico.
- También se puede pedir una comparación entre dos versiones.
- En Subversion, el número de revisión se aplica al estado de todos los ficheros, no a archivos individuales.
- Cada número de revisión es un estado particular del repositorio tras haber sido comitado un cambio.

¿Qué es SVN?

- Conceptos
 - **Trunk**: Tronco, se refiere a una línea base
 - **Brchan**: Rama derivada del tronco
 - **Tag**: Etiqueta

Visual SVN

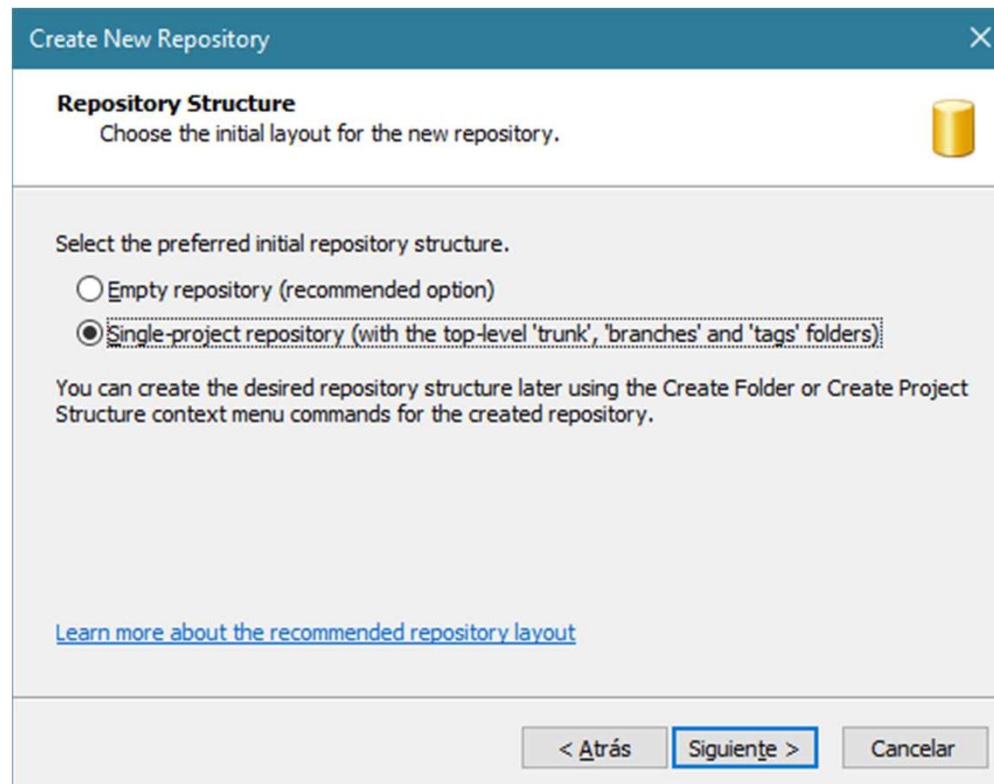
- Herramienta servidor de repositorios SVN.
- Se puede descargar de

<https://www.visualsvn.com/server/download/>

- Dispone de un cliente que permite crear repositorios y usuarios.

Visual SVN

- La instalación es simple. Únicamente marcar la opción **single-project-repository**



Visual SVN

- Esta opción creará automáticamente con el **Commit 1**, los directorios
 - **Trunk**
 - **Branches**
 - **Tags**
- Visual SVN Server, dispone de una consola para manejar SVN.

TortoiseSVN

- Herramienta, que permite manejar de forma grafica e integrada con el SO, los repositorios de SVN.
- Permite hacer **Checkout** de un proyecto a un directorio.
- Permite realizar todas las tareas de SVN.
- Incluye una herramienta visual para mostrar el estado del repositorio, dibujando gráficamente las ramas que lo componen, llamada **Revision Grap**.

Subversive

- Proyecto para la integración de Subversion para Eclipse
 - <http://www.eclipse.org/subversive/>
- Permite trabajar con repositorios SVN de modo similar al que se trabaja con CVS
- Características
 - Exploración de repositorio remoto
 - Compartición de proyectos y check-out
 - Sincronización de proyectos para cambios incoming/outgoing
 - Commit, update y revert
 - Histórico de cambios
 - Resolución de conflictos por mezcla (merge)

Subversive

- Perspectiva SVN Repository Exploring
- Añadir svn://localhost
- Compartir el proyecto con este repositorio
 - Botón derecho > Team > Share project
- Opciones
 - De acceso: Synchronize, Commit, Update, Revert
 - De estructura: Tag, Branch, Merge, Switch
 - Otras: Show properties, Set property, Copy to, Export
- También se puede crear un proyecto como check-out de un repositorio

Subversive



New Repository Location

Enter Repository Location Information

Define the SVN repository location information. You can specify additional settings for proxy and svn+ssh, https connections.

SVN

General | Advanced | SSH Settings | SSL Settings | Proxy

URL:

Label

☒ Use the repository URL as the label

☐ Use a custom label:

Authentication

User:

Password:

☐ Save password

 Saved secret data are stored on your computer in a file that's difficult, but not impossible, for an intruder to read.

Show Credentials For:

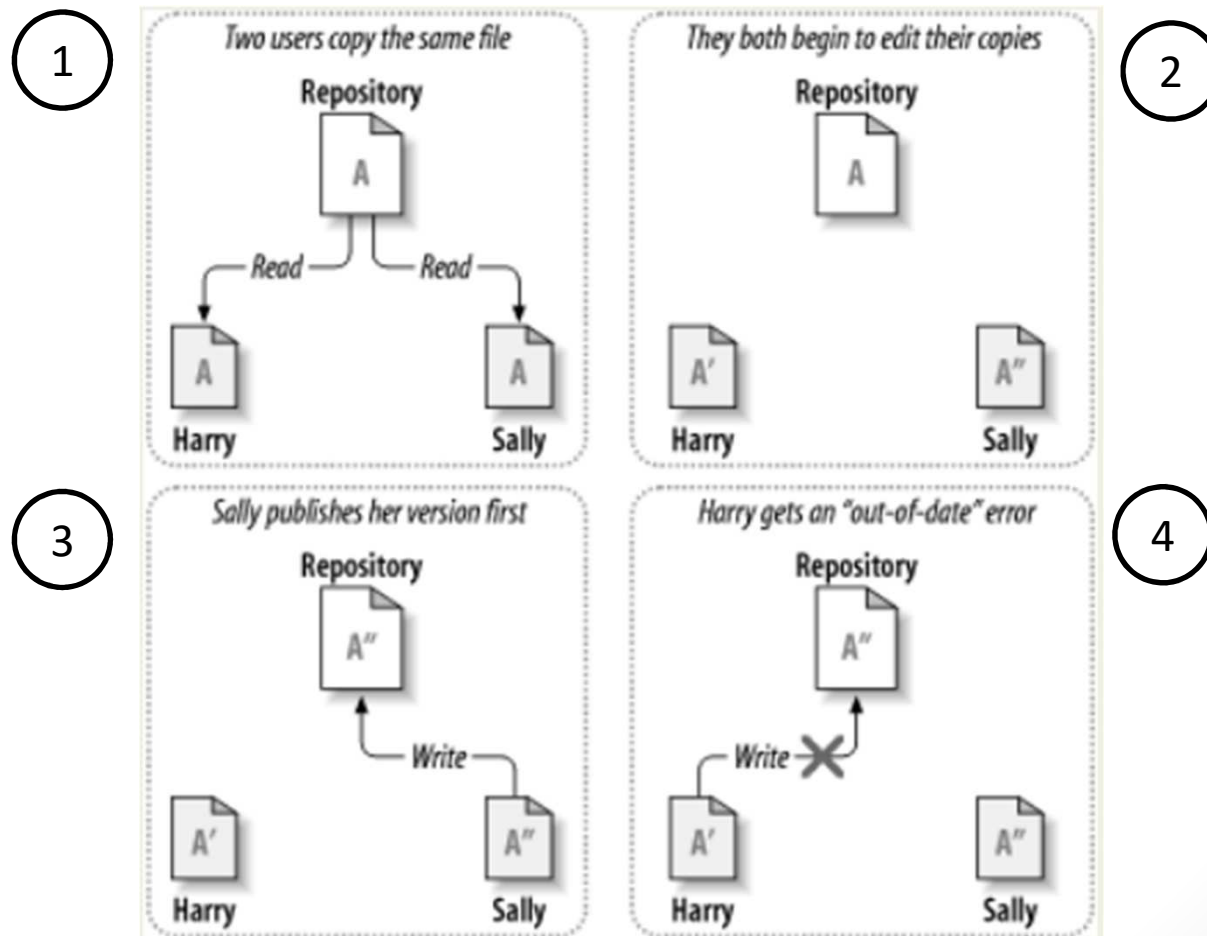
☒ Validate Repository Location on finish

Modo de trabajo

- El modo de trabajo recomendado es **Copiar-Modificar-Mezcar**, que permite
 - Operar con el repositorio sin bloqueos.
 - Si dos usuarios modifican un mismo fichero, el primero en hacer **commit**, no tendrá problemas, al segundo, se le avisará de un conflicto, por lo que tendrá que mezclar sus cambios con las del otro usuario.

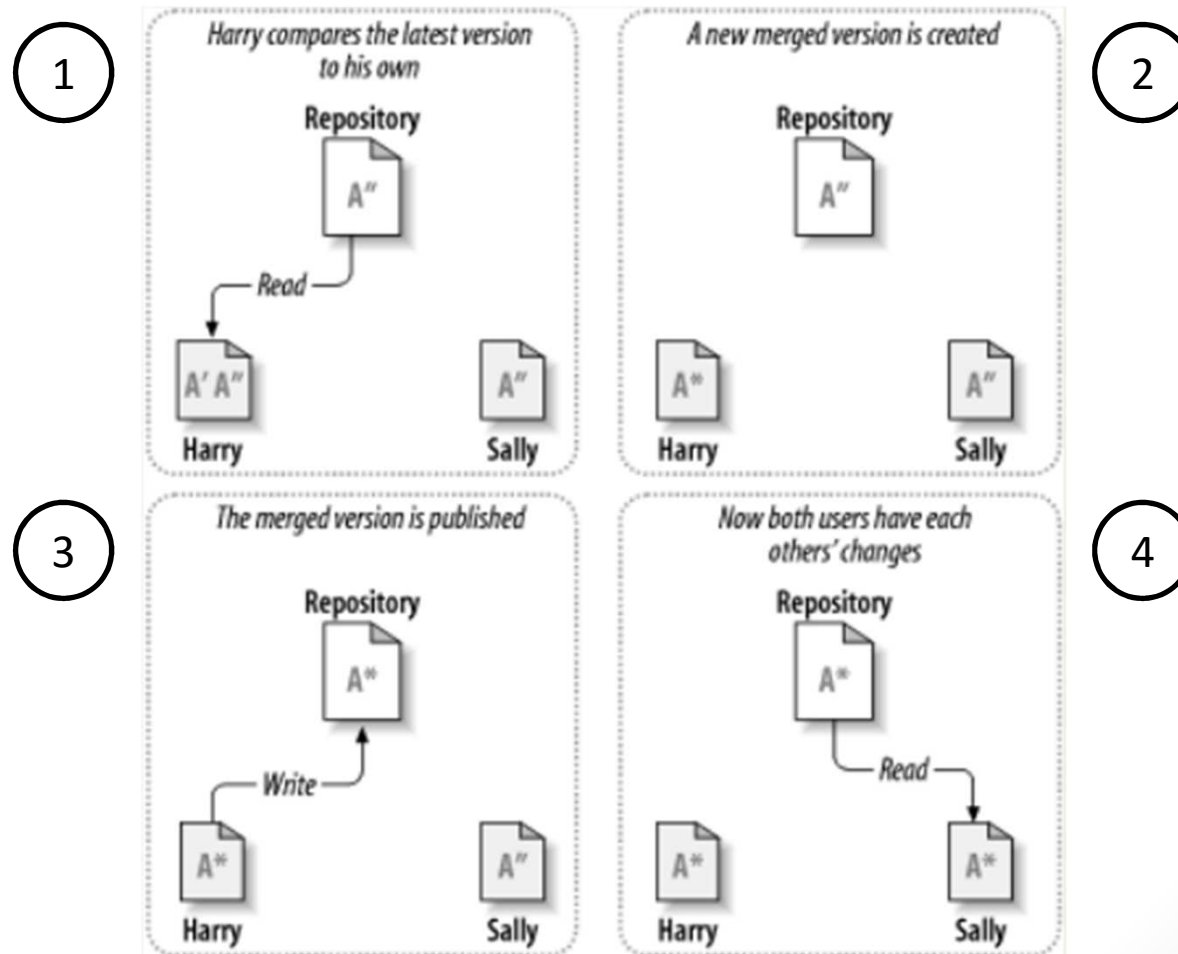
Modo de trabajo

- Escenario



Modo de trabajo

- Procedimiento de trabajo ante conflictos.



Checkout

- Permite descargar una copia de un repositorio remoto a local.

```
svn checkout <url> <path>
```

- También se puede emplear el abreviado **co**.

```
svn co <url> <path>
```

Status

- Se puede conocer el estado de los ficheros que actualmente componen el repositorio local.

```
svn status
```

- Solo se mostrará el estado si no es sincronizado con el repositorio remoto.

Status

- Es estado puede ser
 - **A**: Fichero añadido.
 - **D**: Fichero eliminado.
 - **M**: Fichero modificado.
 - **?**: Fichero local nuevo (no existe en repositorio).
 - **!**: Fichero remoto distinto de la copia local.
 - **U**: Fichero actualizado desde repositorio.
 - **G**: Fichero guardado en repositorio.
 - **C**: Fichero con conflictos.

Add

- Permite añadir elementos al repositorio local.

```
svn add .
```

- Los fichero añadidos al repositorio local, formarán parte del siguiente **commit**.

Revert

- Permite extraer del repositorio local un fichero que previamente ha sido añadido, pero que todavía no sido comitado nunca.

```
svn revert <path>
```

- O también desechar los cambios locales que haya en un fichero ya comitado alguna vez.

Commit

- Añadir cambios en el repositorio remoto.

```
svn commit -m "<mensaje del commit>"
```

- Es obligatorio indicar un mensaje que indique que se ha hecho en el commit.

Log

- Permite mostrar los cambios que se han producido en el repositorio.

```
svn log --diff README.txt
```

- Muestra los cambios en los distintos commit que han afectado al fichero README.txt
- Para mostrar los cambios en el commit actual, que es al que apunta el puntero HEAD.

```
svn log -r HEAD --diff
```

- Para conocer los cambios en un commit.

```
svn log -r <nº de commit> --diff
```


Log

- Para ver todos los cambios que se han producido en el repositorio.

```
svn log -r 1:HEAD --diff
```

- Para ver los cambios entre dos versiones.

```
svn log -r <version inferior>:<version superior> --diff
```

- Para mostrar todos los commit de la rama actual

```
svn log --verbose
```

Update

- Permite sincronizar el repositorio local con el remoto, tanto con el HEAD

```
svn update
```

- Como con cualquier versión.

```
svn update -r <nº de version>
```

- No se pueden hacer commits de versiones intermedias en una rama con commit por arriba, para ello están las ramas.

Branch

- Para crear un “**Branch**”, se ha de copiar una versión del espacio de trabajo “**Trunk**” en la carpeta Branches del repositorio.

```
svn copy <host de la versión en el Trunk> <host del nuevo Branch a crear>  
-m "<Mensaje que defina para que se crea el Branch>"
```

- También se puede emplear el abreviado **cp**.
- A partir de ese momento para trabajar con la rama, se creará otro repositorio local con dicha rama.

Merge

- Cuando se desarrolla sobre un Branch, lo normal es que se lleve alguna versión de dicho Branch sobre el Trunk.

- Para ello

```
svn merge --reintegrate <Url de la rama con la que mezclar>
```

- El repositorio local ha de estar sincronizado, para ello

```
svn update
```

Tags

- Copia del directorio del proyecto, no modificable.
- Se emplea para marcar versiones entregables del proyecto.
- Se crean con

```
svn copy <host de la versión en el Trunk> <host del nuevo Tag a crear>  
-m "<Mensaje que defina para que se crea el Tag>"
```