

Servicio Subversion

Enero 2014

Contenido

- 1. Introducción..... 1
- 2. Acceso al servicio 2
- 3. Carga inicial del repositorio 2
- 4. Creación de la “copia de trabajo” (“working copy”)..... 3
- 5. Modificación de la “copia de trabajo” 3
- 6. Añadir archivos al proyecto 4
- 7. Actualización de la “copia de trabajo” 4
- 8. Enlaces de interés..... 5

1. Introducción

A continuación se mostrará cómo acceder al servicio de repositorios Subversion del RDLab, y como realizar las tareas principales en modo texto (por línea de comandos).

Existen otros clientes de subversion con interficie gráfica que quedan fuera del alcance de esta guía. Algunos de los más populares son [Subclipse \[1\]](#) y [Tortoise \[2\]](#).

En caso de buscar información acerca de la aplicación SVNFrontend, la podréis encontrar en el manual ubicado en:

<http://rdlab.lsi.upc.edu/index.php/es/servicios/documentacion.html>.

2. Acceso al servicio

Para tener acceso al servicio se debe solicitar al RDlab un nombre de usuario (en adelante `<username>`) y una contraseña (`<password>`). Se pueden pedir vía web en <http://rdlab.lsi.upc.edu/index.php/es/servicios/peticionacceso.html> o mediante correo electrónico a rdlab@lsi.upc.edu.

Una vez recibidos, el usuario ya dispone de un repositorio para sus proyectos. Éste se puede ver en <http://svn-rdlab.lsi.upc.edu/subversion/<username>>.

3. Carga inicial del repositorio

Si el proyecto todavía no existe en el repositorio, el primer paso es crearlo y cargar el contenido del proyecto en el repositorio. Para ello hay que crear una carpeta temporal donde se copiará el contenido del proyecto que se quiere poner bajo revisión. Supongamos que se copia en la carpeta “miProyecto-temp” y que contiene un archivo “HolaMundo.java”. Desde esta carpeta se ha de ejecutar el comando “svn import”:

```
$ svn import . http://svn-rdlab.lsi.upc.edu/subversion/  
<username>/miProyecto -m "primer import"  
--username <username>
```

El servidor solicitará contraseña

Clave de '<username>':

Se introduce y el sistema mostrará la siguiente salida:

Añadiendo miProyecto/HolaMundo.java Commit de la revisión 1.

Este proceso ha creado una carpeta “miProyecto” en el repositorio con el contenido del proyecto y con el comentario “primer import”.

Ahora podemos borrar la carpeta temporal que habíamos creado:

```
$ rm -rf miProyecto-temp
```

4. Creación de la “copia de trabajo” (“working copy”)

Si ya existe un proyecto en el repositorio podemos crear una “copia de trabajo”. La “copia de trabajo” es el directorio de nuestro ordenador donde se encuentra la copia del proyecto que se sincronizará con la copia del repositorio.

Hay que destacar que el comando “svn import” descrito anteriormente sólo ha subido el contenido del proyecto al servidor de subversion, pero no ha creado una “copia de trabajo”. Para hacer esto hay que crear una nueva carpeta y ejecutar el comando “svn checkout”:

```
$ cd ..
$ mkdir miProyecto
$ svn checkout http://svn-rdlab.lsi.upc.edu/subversion/
<username>/miProyecto/ miProyecto --username
<username>
```

El sistema nos mostrará la salida:

```
A      miProyecto/HolaMundo.java Revisión obtenida: 1
```

En este momento ya se dispone de una copia asociada al repositorio. Ya se pueden hacer modificaciones en la copia local y subirlos al repositorio (cómo se explica en el siguiente apartado) o actualizar la copia local con cambios que hayan hecho otros usuarios en el repositorio (apartado 7).

5. Modificación de la “copia de trabajo”

Cuando se realizan cambios en la “copia de trabajo” y se desea incorporarlos al repositorio hay que ejecutar el comando “svn commit” desde la carpeta de la “copia de trabajo”:

```
$ cd miProyecto
$ svn commit . -m "modificado HolaMundo" --username
<username>
```

Si, por ejemplo, hemos modificado el archivo HolaMundo.java, el sistema nos mostrará la salida:

```
Enviando      HolaMon.java
Transmitiendo contenido de archivos .
Commit de la revisión 2.
```

6. Añadir archivos al proyecto

Si se crean archivos o carpetas nuevos en la copia de trabajo y se desea incorporarlos al repositorio se deben ejecutar los comandos “svn add” para añadir los archivos al proyecto y posteriormente “svn commit” para cargarlos en el repositorio tal como se explica en el apartado anterior. Si, por ejemplo, se ha añadido el archivo “archivoNuevo.java”:

```
$ svn add archivoNuevo.java --username <username>
```

Y el sistema nos mostrará la salida:

```
A      archivoNuevo.java
```

7. Actualización de la “copia de trabajo”

Si otro usuario ha hecho cambios en el proyecto y los ha incorporado al repositorio, se pueden incorporar dichos cambios a nuestra “copia de trabajo” con el comando “svn update”:

```
$ svn update . --username <username>
```

El sistema nos mostrará la salida:

```
Añadiendo      miProyecto/HolaMundo.java En la revisión 3.
```

Con el comando “svn update” también se puede recuperar una revisión anterior de un archivo o de toda una carpeta con la opción “-r” seguida del número de revisión. Por ejemplo para recuperar la revisión 2 del archivo “HolaMundo.java”:

```
$ svn update ./miProyecto/HolaMundo.java -r2
--username <username>
```

Para recuperar toda la carpeta “miProyecto” en la revisión 2:

```
$ svn update ./miProyecto -r2 --username <username>
```

8. Enlaces de interés

1. Subclipse (subversion para Eclipse):
<http://subclipse.tigris.org/>
2. Tortoise (subversion para Windows):
<http://tortoisesvn.net/>
3. Subversion, página oficial
<http://subversion.apache.org/>
4. Libro “Control de versiones con Subversion” (O'Reilly Media)
<http://svnbook.red-bean.com/>