

TEMA 4-A03: Control de versiones

Índice

1.A03. Control de versiones	3
1.1 Introducción	3
1.2 Actividade	3
Definición	3
Características	3
Funcionamiento repositorio	4
Funcionamiento import	4
Funcionamiento checkout	4
Funcionamiento commit	4
Funcionamiento update	5
Funcionamiento tag	5
Funcionamiento branch	5
Funcionamiento merge	6
Funcionamiento diff	6
Exemplo evolución de un proxecto	6
Clasificación	7
Modelo cliente-servidor	7
Modelo distribuido	7
Programas software libre	7
Modelo cliente-servidor	7
Modelo distribuido	7
Programas software propietario	8
Modelo cliente-servidor	8
Modelo distribuido	8
Subversion	8
Introducción a Subversion	8
Funcionamento	8
Instalación de servidor Subversion CollabNet Subversion Edge	10
Administración básica de servidor Subversion CollabNet Subversion Edge	15
Clientes gráficos de Subversion	20
Control de versiones en netbeans con Subversion	21
Introducción	21
Instalar o cliente Subversion	22
Modificar a variable Path	24
Configurar NetBeans para que localice o software do cliente Subversion	24
Operacións máis comúns con Subversion	24
TortoiseSVN	37
Introducción	37
Instalación de TortoiseSVN	37
Utilización de TortoiseSVN	40
Utilización de Tortoise en local. Create repository here	40
Utilización de Tortoise. Cliente e servidor na mesma máquina	46
Utilización de Tortoise. Cliente e servidor en diferentes máquinas virtuais e mesmo ordenador	47
1.3 Tarefas	48
1.3.1 Tarefa 1. Instalación de servidor Subversion CollabNet Subversion Edge	48
1.3.2 Tarefa 2. Exercicios para practicar con Subversion en NetBeans	48
1.3.3 Tarefa 3. Instalación de TortoiseSVN nunha máquina virtual Windows 7 de 32 bits	49

1.3.4Tarefa 4. Utilización de Tortoise en local. Create repository here.....	49
1.3.5Tarefa 5. Utilización de Tortoise. Cliente e servidor na mesma máquina.....	49

1. A03. Control de versións.

1.1 Introducción

Na actividade que nos ocupa aprenderanse os seguintes conceptos e manexo de destrezas:

- Identificar a estrutura das ferramentas de control de versións.
- Instalar servidor e cliente para control de versións. Administrar basicamente o servidor. Utilizar basicamente o cliente
- Realizar control de versións no contorno de desenvolvemento libre

1.2 Actividade

Definición

Chámase control de versións á xestión dos diversos cambios que se realizan sobre os elementos dalgún produto ou unha configuración do mesmo.

Unha versión, revisión ou edición dun produto, é o estado estable no que se atopa devandito produto nun momento dado do seu desenvolvemento ou modificación.

Os programas de control de versións permiten traballar de forma conxunta a un grupo de persoas no desenvolvemento de proxectos normalmente a través de Internet.

O control de versións leva a cabo principalmente na industria informática para controlar as distintas versións do código fonte pero tamén se aplica noutros ámbitos como documentación, imaxes, sitios web, e en xeral en calquera proxecto colaborativo que requira traballar con equipos de persoas de forma concorrente.

O control de versións de código está integrado no proceso de desenvolvemento de software de moitas empresas sobre todo se teñen máis dun programador traballando no mesmo proxecto.

Características

Os programas de control de versións realizan funcións indispensables durante a vida dun proxecto entre as que destacan:

- Permitir o control dos usuarios que traballarán en paralelo no proxecto:
 - Establecer os usuarios que terán acceso.
 - Asignarlles o tipo de acceso.
- Con relación aos ficheiros:
 - Permitir o almacenamento dos ficheiros.
 - Permitir realizar cambios sobre os ficheiros almacenados: modificar parcialmente un arquivo, borrarlo, cambiarlle o nome, move-lo,...
 - Dispor dun histórico detallado (cambios, data, motivo, usuario,...) das accións realizadas no tempo.
- Con relación ás versións:
 - Etiquetar os arquivos nun punto determinado do desenvolvemento do proxecto para sinalar unha versión estable e así poder identificala posteriormente mediante esa etiqueta ou tag.

- Dispor dun histórico detallado das versións (etiqueta, usuario responsable, data,...).
- Permitir a recuperación de todos ou algún dos arquivos dunha versión.
- Comparar versións tendo unha vista ou informe dos cambios entre elas.
- Con relación ao proxecto, permitir a creación de ramas ou branches, é dicir, bifurcar o proxecto en dous ou máis liñas que poden evolucionar paralelamente por separado. As ramas poden utilizarse para ensaiar novas características de forma independente sen perturbar a liña principal do desenvolvemento. Se as novas características son estables a rama de novo desenvolvemento pode ser fusionada coa rama principal ou tronco.

Exemplo:

Un proxecto no que a versión 1.0 considérase estable e péchase e no que se crea unha rama para a versión 1.1.

Na rama da versión 1.0 pódese seguir traballando na solución de novos erros que aparecen e seguir creando novas versións: 1.01 e posteriores.

Na rama da versión 1.1 pódense traballar de forma independente pero coa posibilidade de utilizar as modificacións que permitiron emendar erros na versión 1.0.

Proxecto para elaborar o manual dunha aplicación. Un departamento dunha empresa pide adaptar algunha das partes do manual á súa forma especial de traballar.

Poderíase abrir unha rama no tronco principal (manual orixinal) para desenvolver o manual específico de forma independente do manual orixinal. Se aparece un cambio que debese de aplicarse ao tronco e a todas as ramas, por exemplo, un erro ortográfico no tronco, o cambio poderíase aplicar a todos eles.

Funcionamiento repositorio

Os sistemas de control de versións teñen un repositorio para cada proxecto, é dicir, un lugar no que se almacenan todos os arquivos do proxecto.

O repositorio ten que crearse e administrarse para que conteña o proxecto (novo ou importado doutro proxecto), teña a estrutura de directorios necesaria, e sexa compartido por un grupo de usuarios autorizados.

O repositorio normalmente reside nun único servidor.

Funcionamiento import

Esta operación permite subir un proxecto non versionado existente no computador do usuario a un repositorio do servidor Subversion por primeira vez.

O proxecto queda versionado para que outros usuarios poidan facer copia.

O proxecto local queda asociado ao repositorio de Subversion.

Funcionamiento checkout

Esta operación permite que un usuario poida crear un directorio de traballo no seu disco duro local cunha copia dunha versión dun repositorio, normalmente a última.

Esta operación faise a primeira vez que se descarga o proxecto do repositorio.

O directorio de traballo queda asociado ao repositorio de Subversion.

Funcionamiento commit

Esta operación permite actualizar o contido dun repositorio cos cambios do directorio de traballo local. Tamén se coñece como publicar ou chek in.

Un usuario pode modificar algún dos arquivos da súa copia local ou crear arquivos nue-

vos. Unha vez que considere finalizados os cambios debe enviar os cambios realizados no seu directorio de traballo ao repositorio central do servidor.

Os cambios deben de ir acompañados de comentarios que os justifiquen para que todos os usuarios estean informados.

Os sistemas de control de versións poden realizar commit:

- Por bloqueo de arquivo na que o sistema bloquea un arquivo cando un usuario está a facer cambios nel de tal modo que os demais usuarios non poden realizar modificacións ata que non finalizase.
- Por xestión de cambios na que calquera usuario ou grupos de usuarios poden modificar ao mesmo tempo un arquivo. Con este sistema poden aparecer conflitos ou posibles inconsistencias que deberán de ser solucionados mediante ferramentas específicas do sistema ou manualmente.

Exemplo de conflito en commit con xestión de cambios:

- Os usuarios X e E traballan sobre o arquivo A.
- O usuario X publica cambios entre as liñas n1 e n2 ao arquivo A.
- O usuario E non descarga do repositorio o arquivo A tras a publicación do usuario X.
- O usuario E realiza cambios entre as liñas n1 e n2 e tenta posteriormente publicar eses cambios. Neste momento o sistema detecta que a copia local sobre a que traballou o usuario E cambiou e é incapaz de realizar os cambios automaticamente. O usuario E debe resolver o conflito combinando os cambios, ou elixindo un deles para descartar o outro.

Para favorecer o traballo en equipo, recoméndase que as tarefas asignadas para o desenvolvemento resólvanse a curto prazo.

Existen ferramentas específicas para a xestión de tarefas en proxectos de desenvolvemento de software como Atlassian Retallo ou Bugzilla.

Funcionamiento update

Esta operación permite actualizar o directorio de traballo (ten que existir) cos cambios (arquivos modificados, directorio novos, arquivos novos, directorios que quedan baleiros no directorio,...) realizados no repositorio desde a última operación update. Tamén se coñece como sincronizar.

A medida que os usuarios van facendo cambios nos arquivos e vanos publicando no repositorio, os directorios de traballo dos outros usuarios quedan desactualizados.

Funcionamiento tag

Esta operación permite pechar unha versión, o que normalmente implica levar unha copia da versión pechada á rama de xestión de versións (branch).

O peche dunha versión permite que nun determinado momento vólvase a versións anteriores. Por exemplo: Detéctase un erro despois de entrega un software; deberíase de arranxar o erro na versión entregada en lugar de na versión actual, polo que é necesario que a versión entregada estea etiquetada.

Funcionamiento branch

Esta operación permite bifurcar o proxecto en ramas que levarán unha evolución paralela do código de tal maneira que en calquera momento poida realizarse unha fusión de cambios entre elas.

Exemplos:

- Ao pór en produción un software, péchase unha versión do mesmo, e pódese crear unha rama evolutiva na que o proxecto seguirá evolucionando e outra correctiva na que se resolverán os erros que poidan xurdir desa versión.
- Crear unha rama para realizar unha modificación sobre a versión que podería facer inestable o tronco.

A xestión das versións en desenvolvemento complicase a medida que se crean ramas.

Deben de considerarse as ramas como de vida limitada, ben polo peche dunha versión ou por fusión de cambios coa rama da que se fixo a ramificación ou por utilizala para crear un proxecto novo.

Funcionamiento merge

Esta operación permite aplicar todos os cambios realizados entre dúas versións nunha rama a outra rama calquera do repositorio.

A operación patch realiza o mesmo que merge pero límtase a modificacións en ficheiros mentres que merge permite directorios e ficheiros.

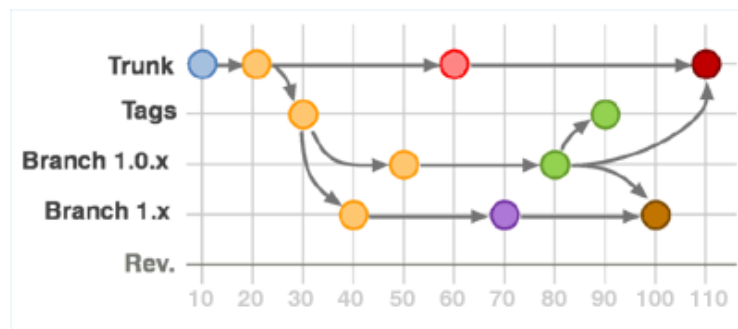
Exemplo:

Supondo que se ten unha rama correctiva e unha evolutiva e que se realizan cambios na rama correctiva para emendar un erro, poderanse fusionar os cambios realizados na rama correctiva cos cambios da rama evolutiva para que así a rama evolutiva non conteña ese erro.

Funcionamiento diff

Antes de enviar os cambios ao repositorio, pode resultar interesante ver os cambios que se realizaron con relación ao repositorio. Esta operación coñécese como diff.

Exemplo ¹ evolución de un proxecto



- Rev. 10: Alta do código fonte no tronco.
- Rev. 20: Evolución do código fonte anterior no tronco mediante o traballo do equipo.
- Rev. 30: Peche do código anterior. Créase a etiqueta para a versión 1.0.0. Ponse en produción a versión 1.0.0. No tronco séguese co desenvolvemento da versión 2.x.
- Rev. 40: Suponse que se contrata un mantemento evolutivo da versión 1.0.0 paralelo ao desenvolvemento da versión 2.0.0, polo que se crea unha nova rama evolutiva para a versión 1.x.

¹ <http://blog.juliopari.com/buenas-practicas-de-gestion-de-versiones-con-subversion/>

- Rev. 50: Suponse que se detecta un erro na versión en produción; para resolvelo hai que recuperar o tag da versión 1.0.0 e para non perder a referencia a esta versión debemos de crear unha rama nova para a versión 1.0.x.
- Rev. 60-80: Trabállase en paralelo no tronco coa versión 2.x, na rama evolutiva para a versión 1.x e na rama correctiva para a versión 1.0.x.
- Rev. 90: Peché do código da versión 1.0.x. Créase a etiqueta para a versión 1.0.1.
- Rev. 100-110: O erro resolto na versión 1.0.1 seguramente estea presente nas versións das ramas 2.x e na da 1.x polo que se debe facer unha fusión de cambios das modificacións realizadas desde a versión 1.0.0 á 1.0.1 co tronco e a rama 1.x.

Clasificación

Programas de control de versións baseados no:

Modelo cliente-servidor

Existe un repositorio centralizado de todo o código nun servidor, cun usuario responsable, ao que acceden os usuarios autorizados mediante un cliente. Facílanse as tarefas administrativas a cambio de reducir flexibilidade, pois todas as decisións fortes (como crear unha nova rama) necesitan a aprobación do responsable.

Modelo distribuido

Cada usuario traballa directamente co seu repositorio local que fai ao mesmo tempo de cliente e de servidor. Non é necesario tomar decisións de forma centralizada. Os distintos repositorios poden intercambiar e mesturar revisións entre eles.

Programas software libre

Modelo cliente-servidor

- CVS (Concurrent Version System <http://www.cvshome.org/>): Desenvolvido por GNU. Distribúese baixo licenza GPL.
- SVN (Subversion <http://subversion.apache.org/>): Hoxe é o máis popular. Foi creado para mellorar CVS sobre todo no manexo de arquivos binarios. Distribúese baixo licenza Apache/BSD.

Modelo distribuido

- Git (<http://git-scm.com/>): Deseñado por Linus Torvalds, está baseado en BitKeeper e Monotone. Distribúese baixo licenza GNU GPL v2. É usado para o proxecto de programación do kernel de Linux.
- Mercurial (<https://www.mercurial-scm.org/>): Creado por Matt Mackall. Distribúese baixo licenza GNU GPL.
- Bazaar (<http://wiki.bazaar.canonical.com/>): Desenvolvido por Canonical LTF e comunidade. Facilita proxectos de software libre e opensource. Distribúese baixo licenza GNU GPL v2.

Programas software propietario

Modelo cliente-servidor

- Visual SourceSafe : Ferramenta de control de versións de Microsoft orientada a equipos pequenos que se integra na contorna de traballo de Visual Studio e no resto de ferramentas de desenvolvemento de Microsoft. Actualmente está a ser substituído por Visual Studio Team Foundation Server (<http://www.visualstudio.com/es-es>).

Modelo distribuído

- BitKeeper (<http://www.bitkeeper.com/>) producido por Bitmover Inc.

Subversion

Introducción a Subversion

CollabNet (<http://www.collab.net>) lanzou o proxecto Subversion a principios do ano 2000 para crear un sistema de control de versións que substituíra a CVS pero sen os seus fallos e defectos. O 31 de agosto de 2001 empezouse a usar Subversion.

Actualmente (2014) o proxecto Subversion (<https://projects.apache.org/project.html?subversion>) é desenvolvido como un proxecto da Software Apache Foundation (<http://www.apache.org/>) e distribúese baixo unha licenza de tipo Apache/BSD.

CollabNet sigue estando involucrada no proxecto Subversion, ofrece CollabNetSubversion baixo licenza GNU Affero General Public License 3.0 (AGPLv3) e dá servizos de soporte, formación e consultoría.

Subversion é un sistema de control de versións utilizado en moitos proxectos. No ano 2013 era utilizado en todo o mundo polo 50% dos equipos de desenvolvedores e por 5 millóns de desenvolvedores.

As principais melloras con respecto a CVS son:

- Ademais de permitir control de versións por arquivo, permiten control de versións por directorio.
- Permite mover arquivos no repositorio.
- Dispón de soporte para conexións seguras.
- A comparación entre arquivos é máis efectiva e rápida.



Descubrir a versión máis actual de Subversion.

Funcionamento

Para o funcionamento de Subversion:

- Ten que existir un servidor Subversion no que estean creados os repositorios e os usuarios cos permisos correspondentes. O servidor Subversion pode manexarse con ferramentas na liña de comandos ou mediante algunha ferramenta gráfica que utiliza esas ferramentas.
- O usuario ten que ter instalado a ferramenta svn (cliente Subversion) no seu equipo local e pode ter instalado un cliente gráfico que utiliza esa ferramenta. Os clientes gráficos máis populares son: Tortoise para Windows e RabbitVCS para Linux. Algúns IDE como NetBeans dispón dun complemento para ser cliente gráfico de Subversion.

As principais ferramentas de Subversion son:

- **svn**

Cliente de Subversion para realizar as operacións principais como obter copia para o directorio de traballo local, sincronizar copia co servidor ou enviar os cambios ao repositorio (libro dixital sobre Control de versiones con Subversion en <http://svnbook.red-bean.com/nightly/es/svn-ch-2.html>)

- **svnadmin**

Administrador de Subversion para creación e administración de repositorios.

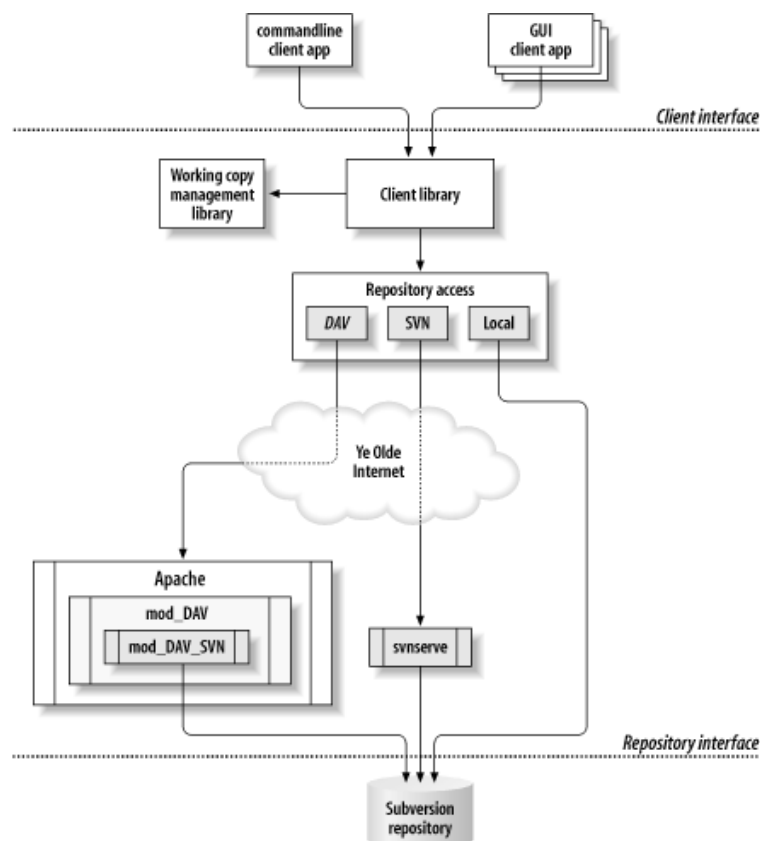
- **svnserver**

Servidor lixeiro que funciona como demo normalmente no porto 3690 e que permite aos clientes svn conectarse co servidor Subversion mediante conexións TCP/IP a través dos protocolos `svn://` ou `svn+ssh://`

- **svnlook**

Administrador para examinar o histórico do repositorio.

O acceso aos repositorios pode facerse localmente no servidor, ou a través de Internet: coa ferramenta `svn` ou mediante un servidor Apache configurado para servir repositorios de Subversion (`mod_DAV_SVN`).



A estrutura de carpetas por defecto dos repositorios de Subversion é:

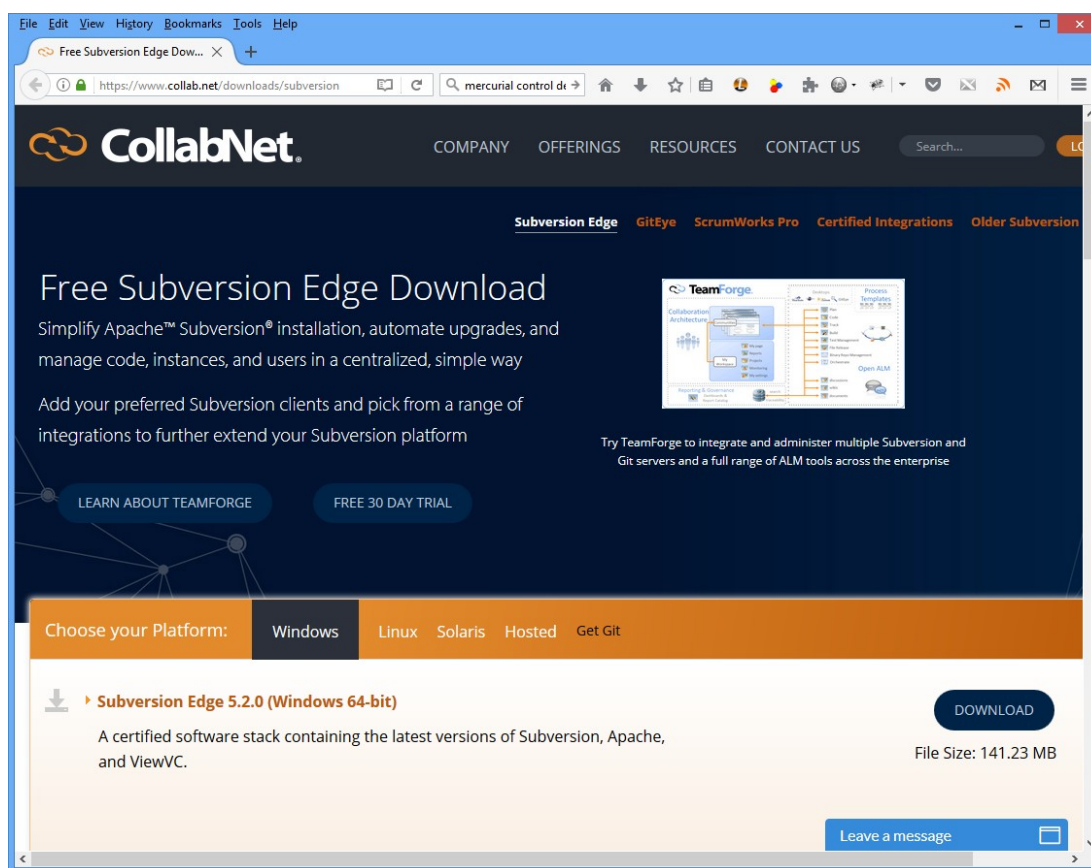
- **/trunk**: Onde se aloxa o tronco ou rama de desenvolvemento principal.
- **/tag**: Onde se aloxarán as versións estables do produto en desenvolvemento, polo tanto non se desenvolverá sobre esta rama.
- **/branch**: Onde se aloxan as ramas de evolución paralelas ao tronco.

Instalación de servidor Subversion CollabNet Subversion Edge

A práctica de instalación do servidor Subversion farase sobre unha máquina virtual con sistema operativo Windows 7 de 32 bits. Utilizarase CollabNet Subversion Edge xa que:

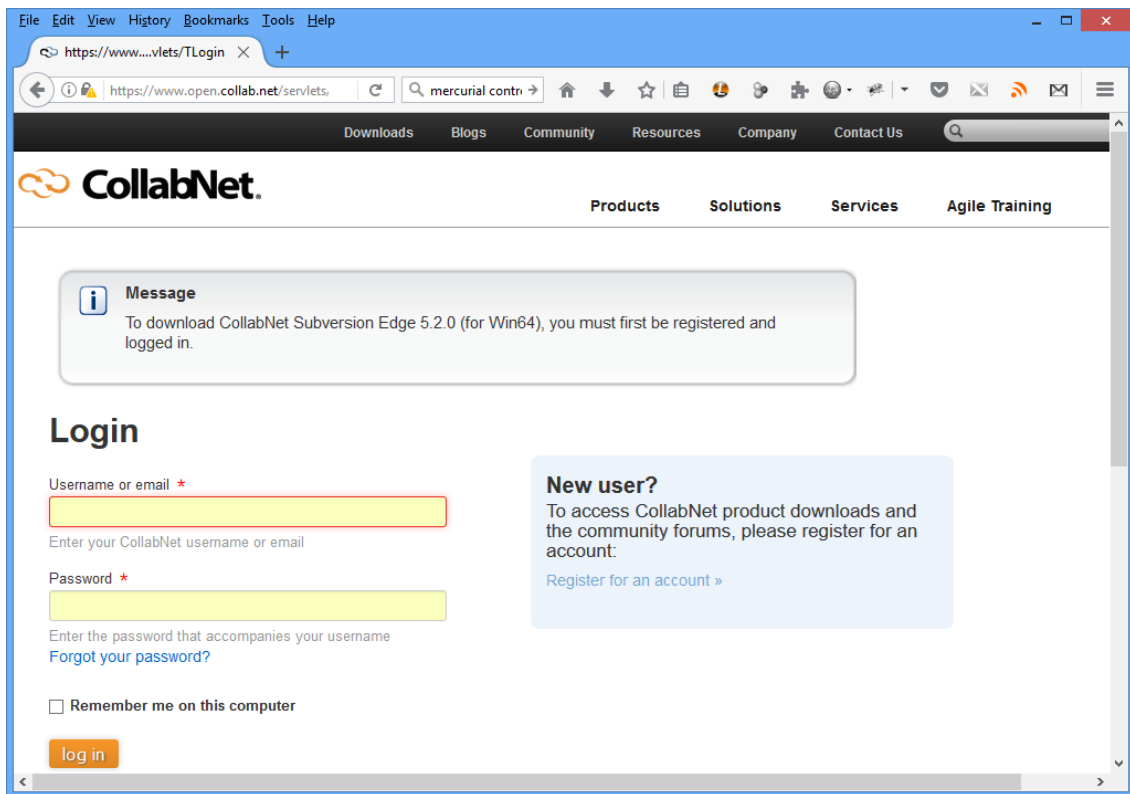
- Inclúe os binarios de Subversion e Apache necesarios para instalar, xestionar e manexar un servidor Subversion.
- Inclúe a ferramenta ViewVC que permite navegar polo repositorio dende un navegador web.
- É moi sinxelo configurar e administrar o servidor Apache e os repositorios Subversion mediante unha potente consola web.
- Facilmente pódese configurar o servidor para utilizar conexións SSL (protocolo criptográfico que proporciona comunicacións seguras por unha rede) e para conectar o servidor a un repositorio LDAP corporativo incluíndo Microsoft Active Directory.
- CollabNet Subversion Edge só admite servir repositorios de Subversion a través de Apache httpd.

Pode descargarse de <http://www.collab.net/downloads/subversion>

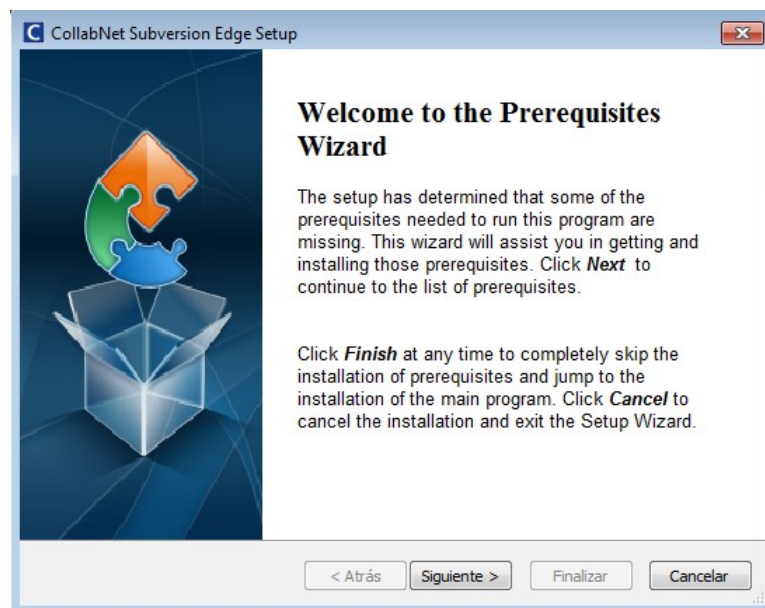


Descárgase a última versión, por exemplo para 64 bits

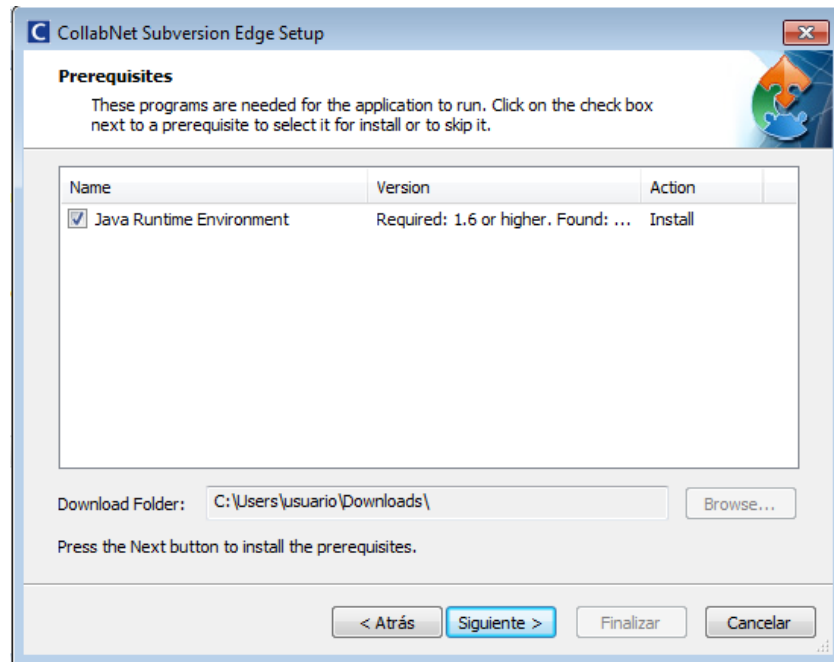
Para descargar o software é necesario rexistrarse creando unha conta (Paso 1); recibirase un correo electrónico cun enlace dende o que se pode poñer o password da conta; con ela poderá identificarse como usuario (Paso 2).



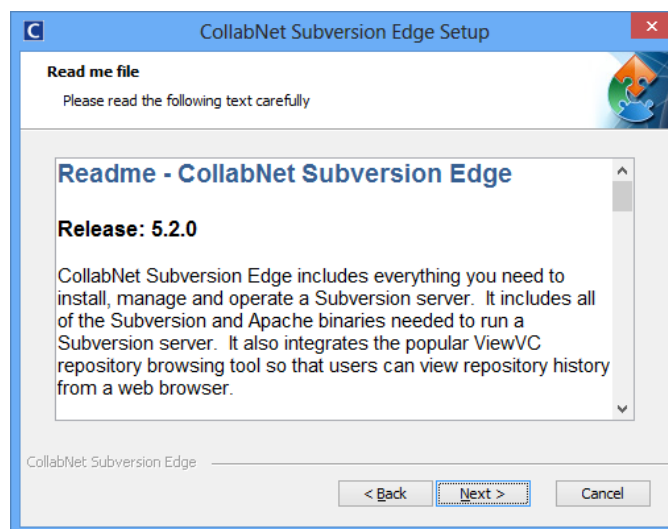
Execútase o arquivo baixado, prémese en *Siguiente* >



Necesita Java 1.6 ou superior. Se o instalador detecta que non existe e sempre que a máquina teña acceso a internet, pode marcarse para instalar e instalarse por defecto en *C:\ProgramFiles\Java\jre6* (pode cambiarse esta localización)



Finalizada a instalación do software dos prerequisites, vólvese a lanzar o *setup* e aparece *Readme* que deberá de lerse.

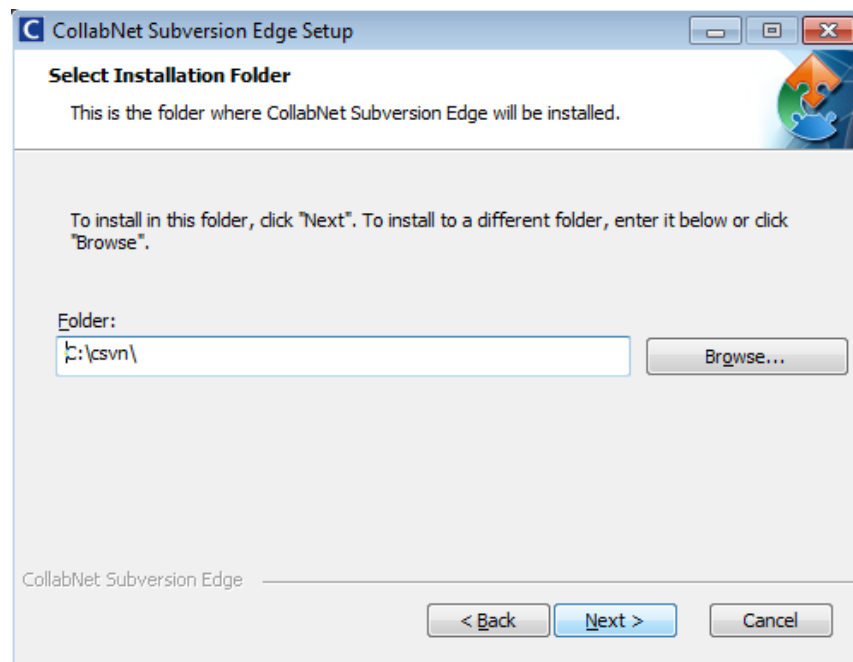


O arquivo *Readme* informa de que o proceso de instalación realizará as seguintes operacións:

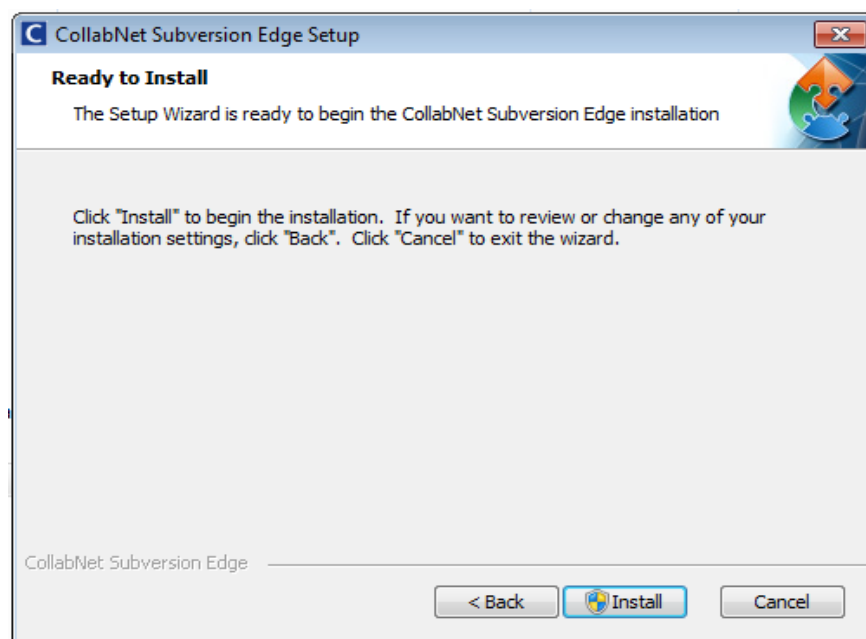
- Copiar os arquivos de CollabNet Subversion Edge.
- Actualizar a variable PATH do sistema para que inclúa o path as carpetas bin e Python25 e creará e actualizará a variable de entorno PYTHONPATH.
- Modificar o firewall de windows para que permita acceder aos binarios de Apache e abrir os portos 3343 e 4434.
- Engadir dous servizos windows para que se inicien cando se inicia o sistema:
 - CollabNet Subversion Edge
 - CollabNet Subversion Server
- Crear un usuario administrador cos valores por defecto:
 - Address: <http://localhost:3343/csvn>

- Username: admin
- Password: admin

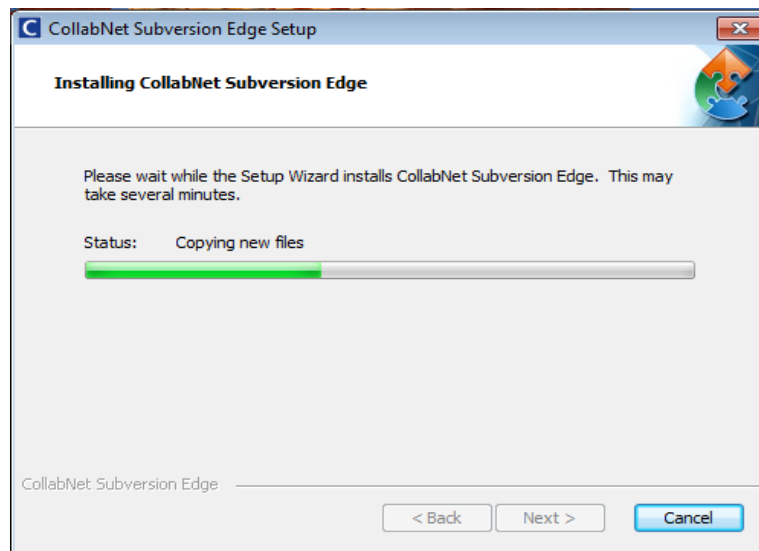
Deixase a carpeta de instalación por defecto.



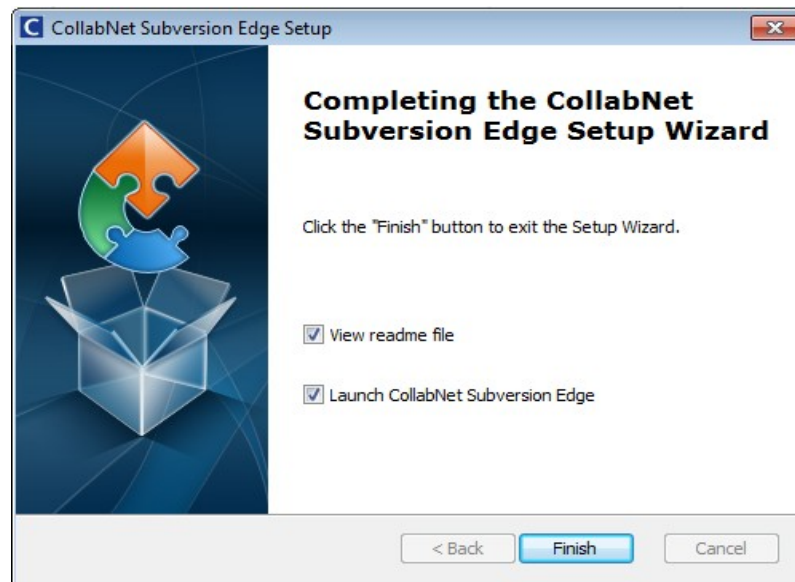
Decídese facer a instalación.



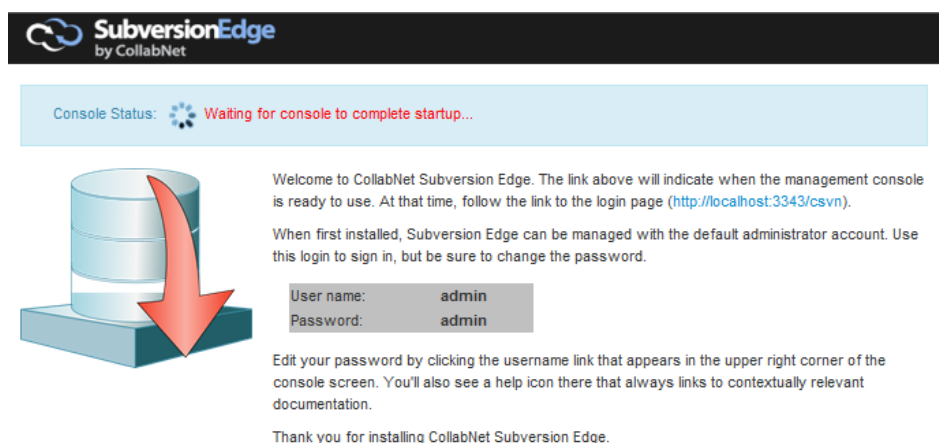
Faise a instalación.



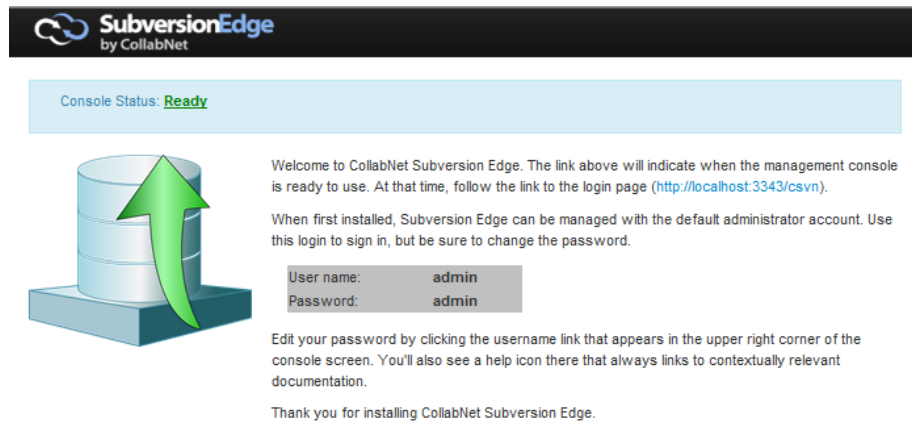
Finaliza a instalación



Se na pantalla anterior se indica que se lance CollabNet Subversion Edge, móstrase unha web local onde aparece un enlace para a consola de administración e informa do usuario e contrasinal para entrar.



Ao finalizar o proceso anterior:

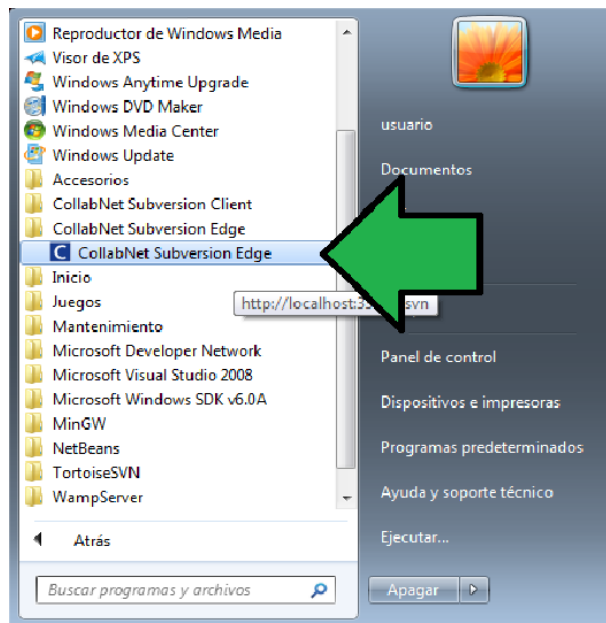


Tarefa 1. Instalación de servidor Subversion CollabNet Subversion Edge

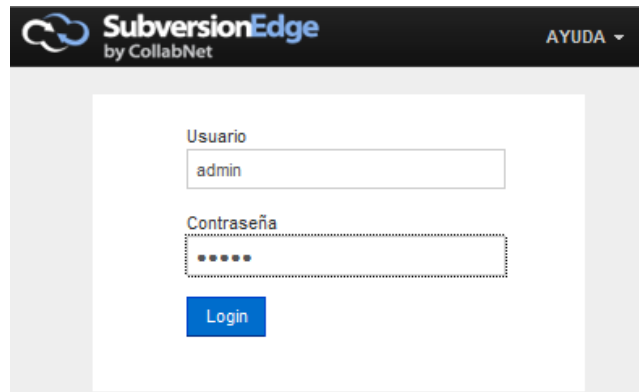
Administración básica de servidor Subversion CollabNet Subversion Edge

A administración de Subversion está accesible:

- Mediante un navegador e a url local anterior: <http://localhost:3343/csvn>
- Ou dende o menú *Inicio*.



Para realizar labores de administración, hai que identificarse como usuario admin:



SubversionEdge
by CollabNet

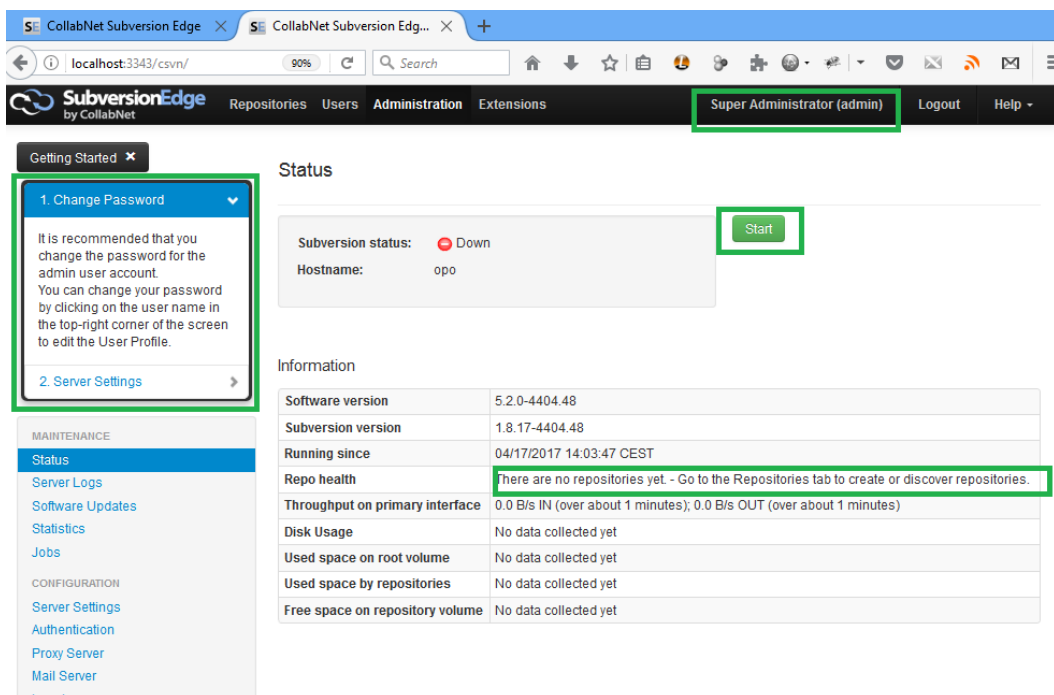
AYUDA ▾

Usuario
admin

Contraseña
•••••

Login

Aparece a páxina web principal de administración na pestana Administración, co menú Estado seleccionado, suxírese o cambio da contrasinal do administrador, informa de que o servidor está apagado e de que non ten repositorios.



CollabNet Subversion Edge

Repositories Users Administration Extensions Super Administrator (admin) Logout Help ▾

Getting Started ✕

1. Change Password ▾
It is recommended that you change the password for the admin user account. You can change your password by clicking on the user name in the top-right corner of the screen to edit the User Profile.

2. Server Settings >

MAINTENANCE

Status

Server Logs

Software Updates

Statistics

Jobs

CONFIGURATION

Server Settings

Authentication

Proxy Server

Mail Server

Logging

Status

Subversion status: ● Down

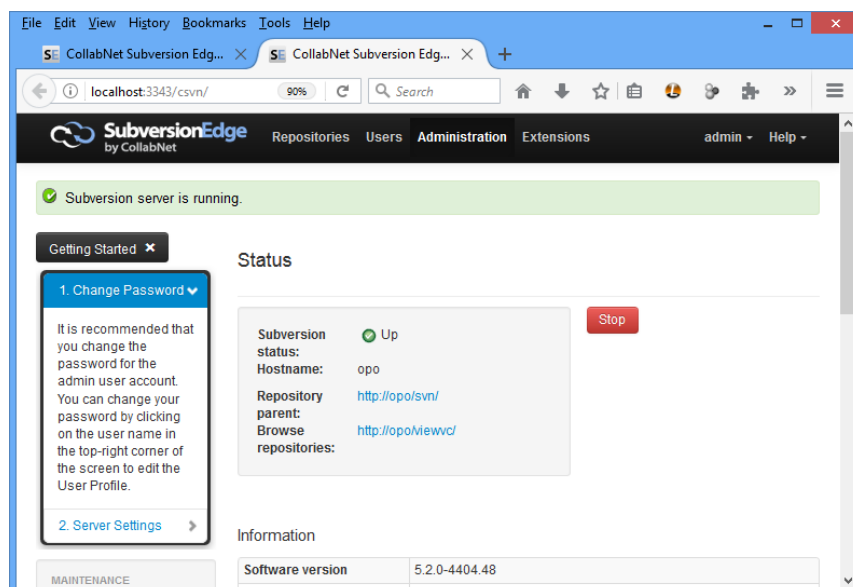
Hostname: opo

Start

Information

Software version	5.2.0-4404.48
Subversion version	1.8.17-4404.48
Running since	04/17/2017 14:03:47 CEST
Repo health	There are no repositories yet - Go to the Repositories tab to create or discover repositories.
Throughput on primary interface	0.0 B/s IN (over about 1 minutes); 0.0 B/s OUT (over about 1 minutes)
Disk Usage	No data collected yet
Used space on root volume	No data collected yet
Used space by repositories	No data collected yet
Free space on repository volume	No data collected yet

Na pestana Administración ponse a funcionar o servidor.



File Edit View History Bookmarks Tools Help

CollabNet Subversion Edge

Repositories Users Administration Extensions admin Help ▾

Subversion server is running.

Getting Started ✕

1. Change Password ▾
It is recommended that you change the password for the admin user account. You can change your password by clicking on the user name in the top-right corner of the screen to edit the User Profile.

2. Server Settings >

MAINTENANCE

Status

Subversion status: ● Up

Hostname: opo

Repository parent: <http://opo/svn/>

Browse repositories: <http://opo/viewvc/>

Stop

Information

Software version	5.2.0-4404.48
------------------	---------------

Na pestana Usuarios pódese xestionar os usuarios.


Pódense establecer roles para a administración do servidor Subversion e non teñen que ver cos permisos de acceso aos repositorios.

Create User

Login Name:	<input type="text" value="probas"/>
Full Name:	<input type="text" value="probas para CODE"/>
New Password:	<input type="password" value="••••••"/>
Confirm Password:	<input type="password" value="••••••"/>
Email:	<input type="text"/>
Description:	<input type="text"/>
Roles Granted:	<div><input type="checkbox"/> ROLE_ADMIN - Super/Root Administrador (Todos los privilegios) <input type="checkbox"/> ROLE_ADMIN_HOOKS - Repository Hook Scripts Administrator <input type="checkbox"/> ROLE_ADMIN_REPO - Administrador del Repositorio <input type="checkbox"/> ROLE_ADMIN_SYSTEM - Administrador del Sistema/Servidor <input type="checkbox"/> ROLE_ADMIN_USERS - Administrador de Cuentas de Usuarios <input checked="" type="checkbox"/> ROLE_USER - Autoridad de Usuario Básico, requerido para acceso desde la consola</div>

Create

Están creados os dous usuarios.

 SubversionEdge
by CollabNet

Repositorios

Usuarios

Administración

Extensions

Super Administrator (admin)

SALIR

AYUDA

USUARIOS

Lista de Usuarios

Lista de Roles

np: You only need to create user accounts for built-in users. If using LDAP, your users are created and managed by your LDAP server, such as Windows Active Directory. Users can login to the console with their LDAP credentials and that will create a "user placeholder" account to which you can assign administrator roles if desired.

Usuarios

10 records per page

Filter:

Usuario	Nombre Completo	Descripción
admin	Super Administrator	admin user
invitado	invitado	usuario de só lectura
probas	probas para CODE	usuario para probas no servidor Subversion

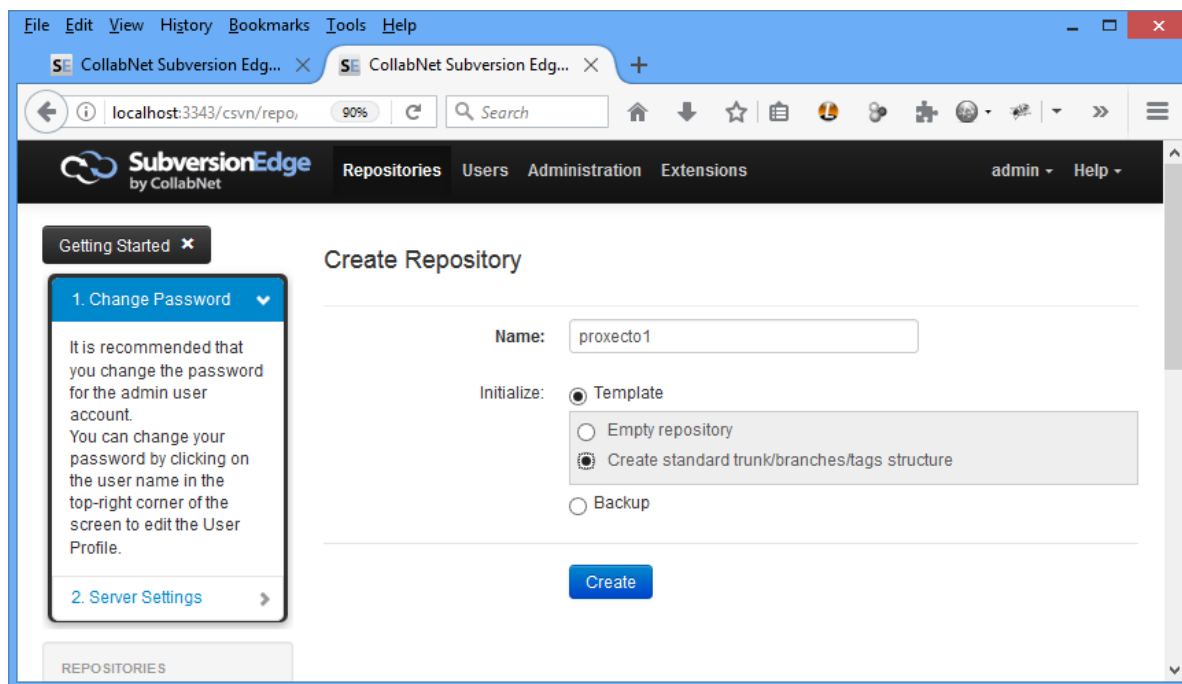
Showing 1 to 3 of 3 entries

Anterior 1 Siguiente

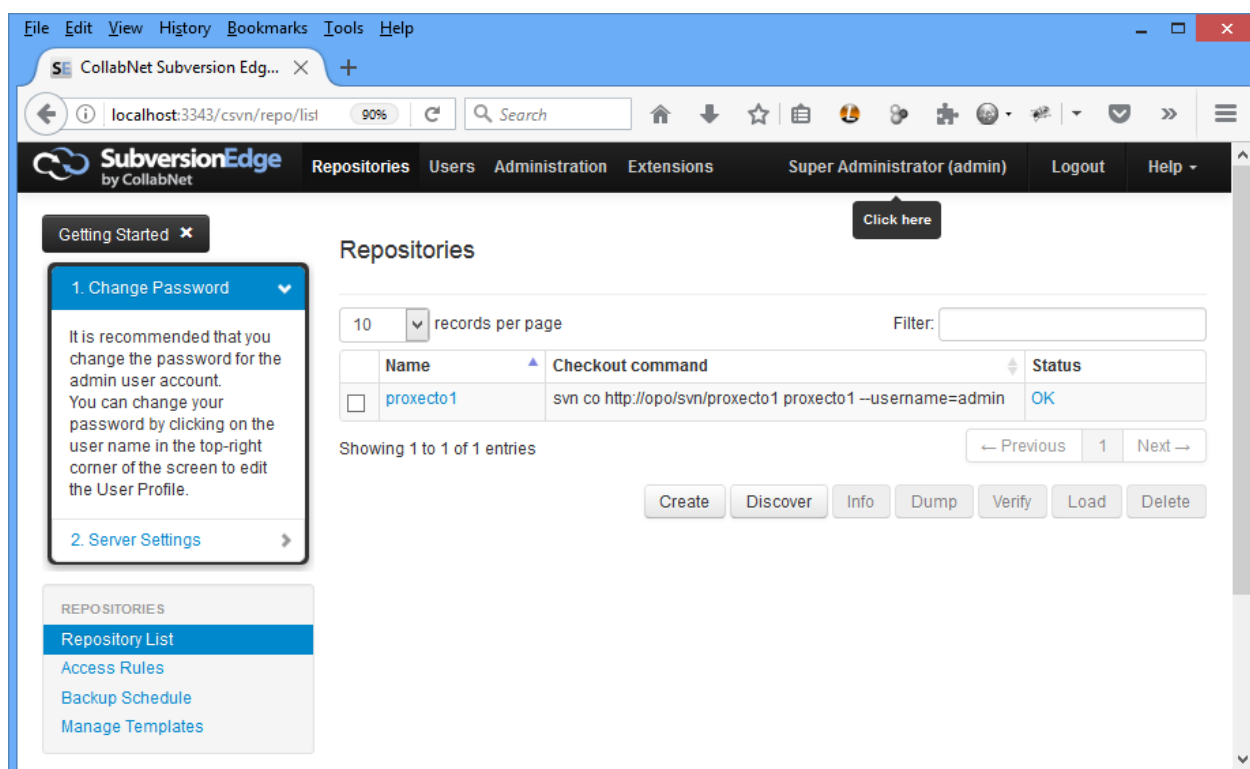
Crear

Na pestana Repositorios pódense administrar os repositorios.

Crear o repositorio proxecto1 coa estrutura de directorios por defecto.

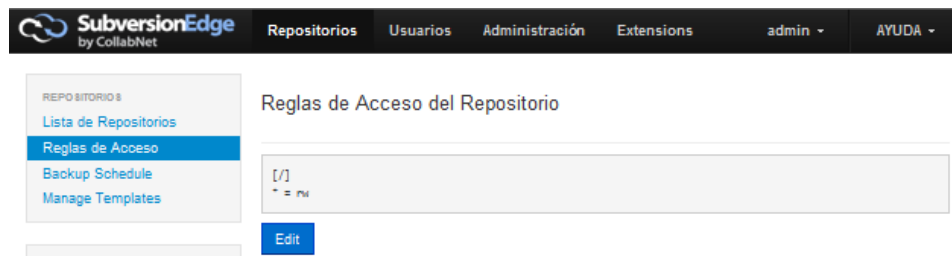


Repositorio creado sen erros e incluído na lista de repositorios. Pode verse o comando svn necesario para facer checkout.



Na pestana Repositorios, opción Reglas de acceso pódense administrar os permisos de acceso dos usuarios aos repositorios ou a unha carpeta dun repositorio editando as regras de acceso.

Por defecto trae:



Que se debe de modificar pois significa que todos os usuarios poden acceder a todos os repositorios con permisos de lectura e escritura.

Para definir permiso de lectura en todo o repositorio chamado “aaaaa”:

[aaaaa:/]

* = r

Para que o usuario carlos poda ler e escribir na carpeta “ccc” do repositorio aaaaa e o usuario pedro só poda ler:

[aaaaa:/ccc/]

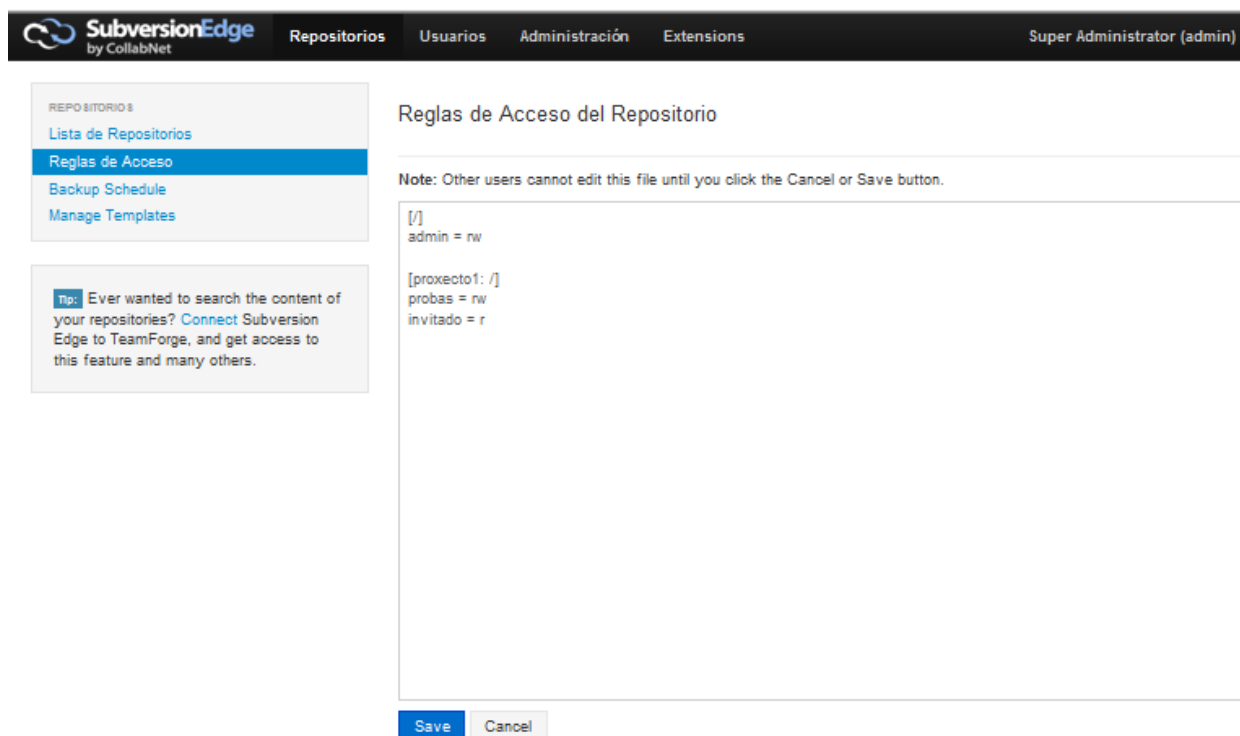
pedro = r

carlos = rw


Deste xeito pódense establecer permisos diferentes para cada carpeta dun repositorio.

Outra posibilidade é facer grupos de usuarios cando os usuarios son moi numerosos para establecer os permisos de xeito mais cómodo. Pode verse información detallada na pestana *Ayuda*.

Para que o usuario admin poida ler e escribir en tódolos repositorios, o usuario probas poida ler e escribir sobre o repositorio proxecto1 e o usuario invitado só poida ler:



Na pestana Repositorios, na opción Lista de Repositorios, pode facerse clic sobre o nome do repositorio proxecto1 para ver o contido. O servidor pedirá que se identifique un usuario autorizado (por exemplo invitado) e pode verse que están creadas as carpetas por defecto pero sen contido.


SubversionEdge
 by CollabNet

[Roots](#)
[Changeset](#)
[Change Log](#)
[Directory](#)

Logged in as: invitado
 [Help](#)

/

File	Rev.	Age	Author	Last log entry
branches/	1	28 minutes	SYSTEM	Creating_initial_branch_structure
tags/	1	28 minutes	SYSTEM	Creating_initial_branch_structure
trunk/	1	28 minutes	SYSTEM	Creating_initial_branch_structure

Files shown: 0
 Directory revision: 1 (of 1)
 Sticky revision: [Get Sticky Revision](#)

Accessing the repository

```
svn checkout http://PCProbas/svn/proxectol/trunk proxectol --username invitado
```

Cientes gráficos de Subversion

TortoiseSVN.



TortoiseSVN é un cliente de Subversión, implementado como extensión do shell de Windows e que é software libre.

Con TortoiseSVN é realmente fácil controlar as versións e modificacións de arquivos ou de código fonte do software en entorno Windows.

Posto que non é unha integración para un IDE específico pódese usar con calquera ferramenta de desenvolvemento.

RabbitVCS



RabbitVCS é un programa en entorno gráfico para sistemas de control de versións en Linux. Intégrase no explorador de arquivos para proporcionar acceso nun menú contextual ós ficheiros e control de versións nos repositorios.

O proxecto era orixinalmente chamado NautilusSvn, debido a integración co sistema de arquivos Nautilus, foi rebautizado a RabbitVCS (Sistema de Control da Versión).

A interface foi inspirada por TortoiseSVN, recoñecible pola integración no administrador de arquivos, dispoñíbel en Nautilus e Thunar. Tamén se pode integrar con Gedit ou executar independentemente na liña de comandos.



NetBeans IDE proporciona unha estreita integración co cliente Subversion.

O soporte Subversion do IDE foi deseñado para simplificar o proceso de compartir un repositorio entre un grupo de traballo e realizar labores de control de versións directamente dende o proxecto do IDE.

Para poder utilizar NetBeans como cliente Subversion hai que:

Ter instalado o software cliente de Subversion para que NetBeans poida executar os comandos svn.

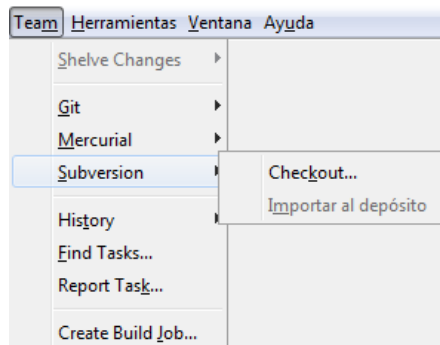
Configurar NetBeans para que coñeza a ruta do cliente Subversion.

Control de versións en netbeans con Subversion

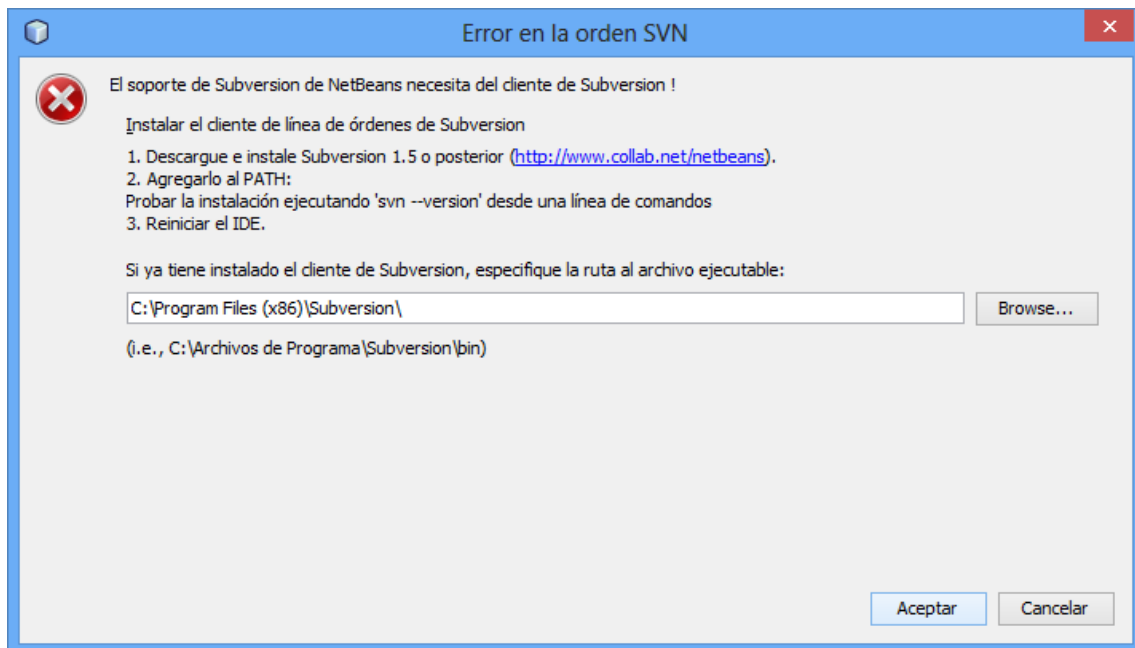
Introdución

NetBeans dispón dun complemento para poder utilizarse como cliente de Subversion. Os pasos a seguir para que NetBeans funcione como cliente Subversion son:

- Instalar o software cliente de Subversion.
- Configurar NetBeans para que localice o software do cliente Subversion.
- Asegurarse de ter un repositorio no servidor Subversion para poder utilizalo.

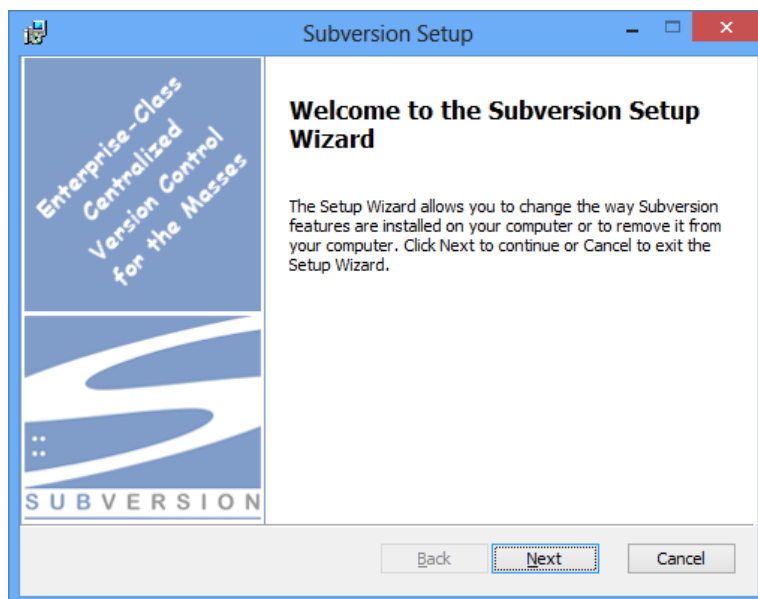


Se pretende utilizar o control de versións con Subversion sen ter instalado o software cliente de Subversion, aparecerá unha ventá coa mensaxe de erro indicando que non se poden executar as ordes svn e suxerindo opcións para a instalación como a que segue:

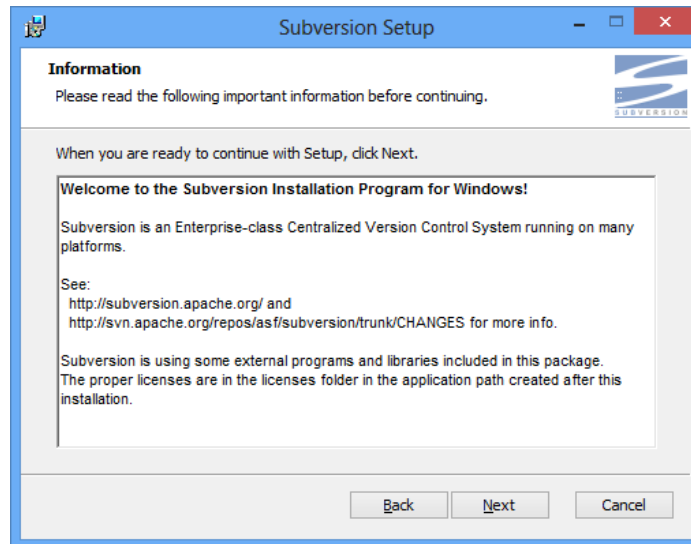


Instalar o cliente Subversion

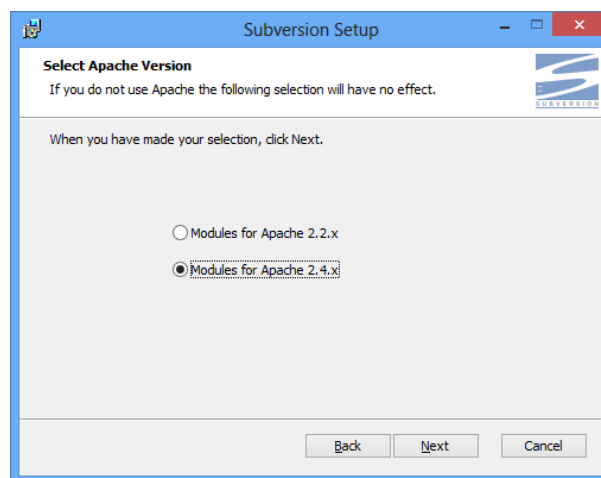
Unha forma é dende a ventá de erro anterior, na que se elixe a opción recomendada e prémese en Aceptar ou ben se vai a <https://www.collab.net/downloads/subversion>



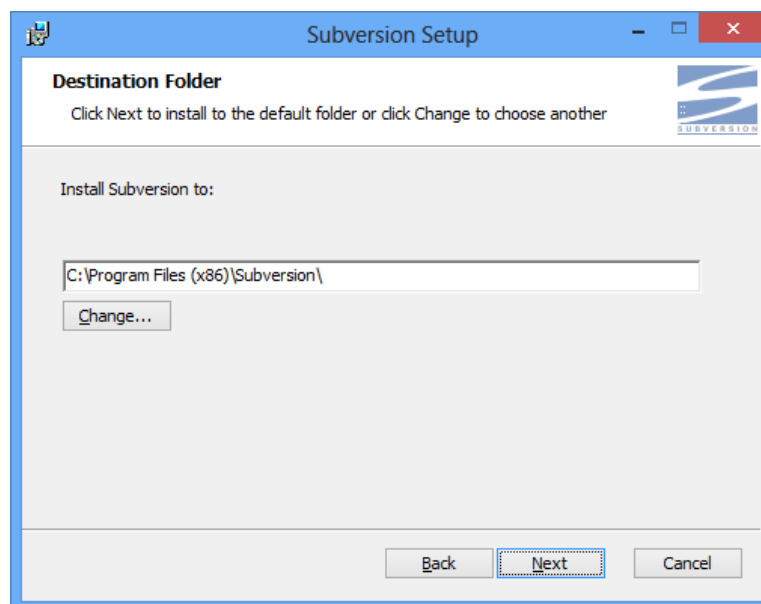
Darlle ó botón Next e aparecerá unha ventá con Información sobre a Instalación.



Hai que escoller a versión dos módulos de Apache. No noso caso o 2.4x



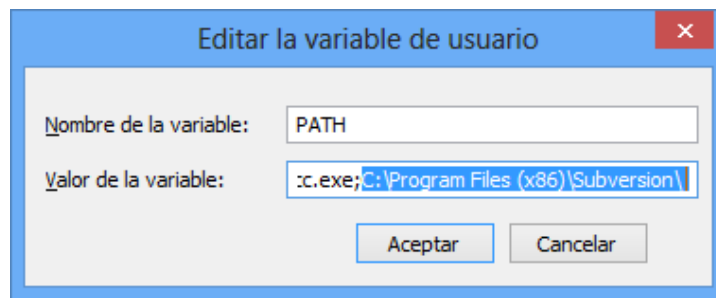
Comprobar o directorio de instalación, que será o que haberá que engadir en Netbeans



E por último darlle a Install para comezar o proceso de instalación

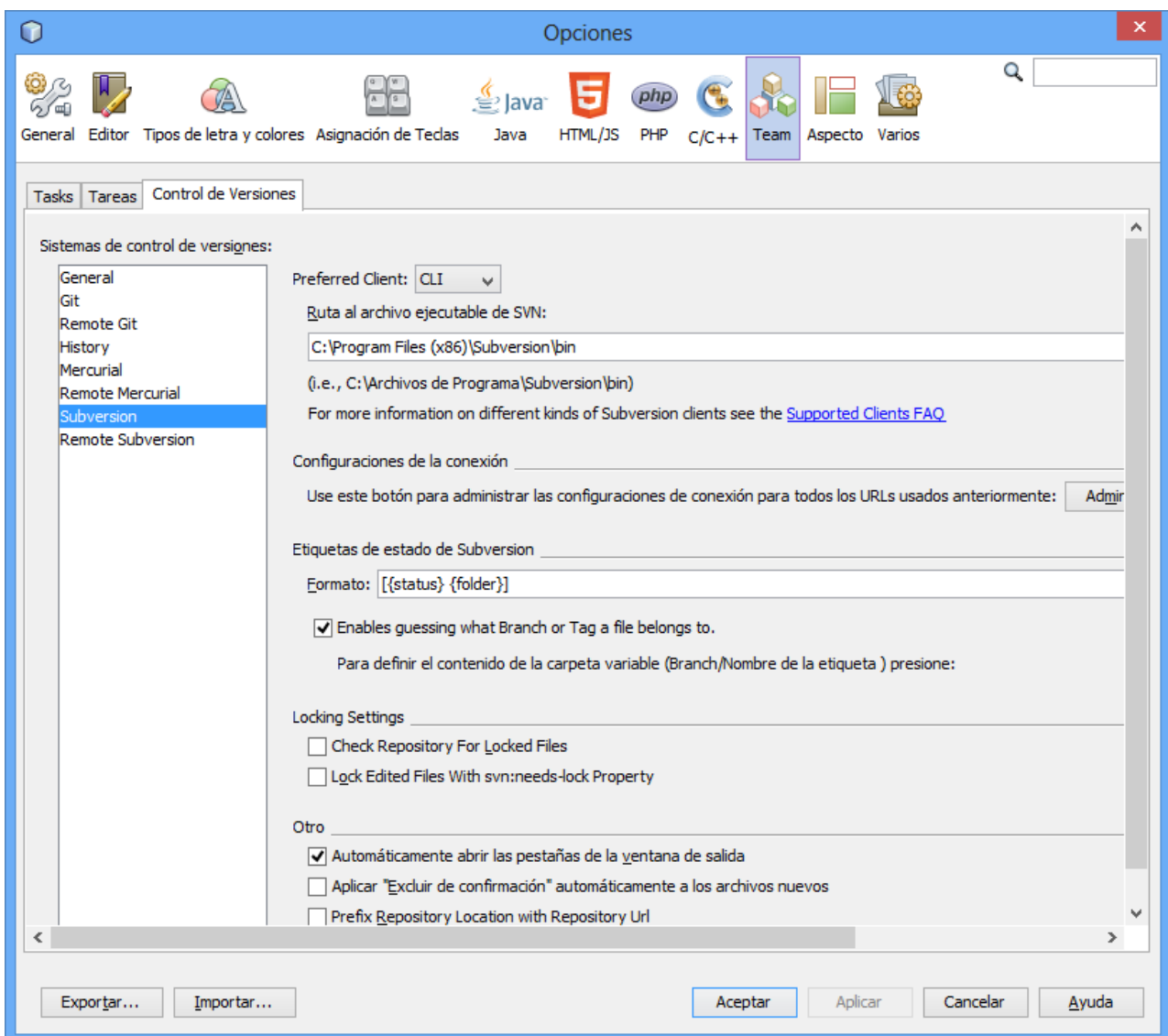
Modificar a variable Path

Ir a Equipo->Propiedades-> Configuración avanzada do sistema-> Pestana de Opcións avanzadas-> Botón "Variables de Entorno" e na variable Path engadir a ruta de instalación de Subversion separando por ; os valores anteriores



Configurar NetBeans para que localice o software do cliente Subversion

Hai que elixir Herramientas->Opciones->Team->Control de versiones->Subversion e examinar para indicar a ruta do arquivo svn.exe.



Operacións máis comúns con Subversion

As operacións máis comúns son:

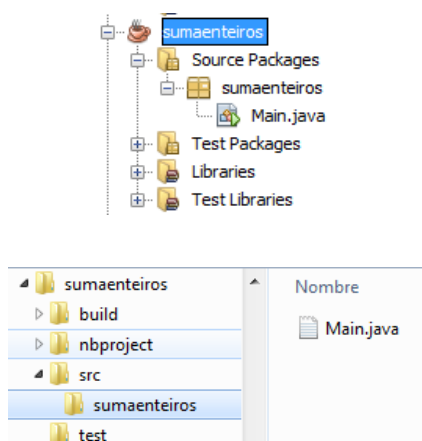
- Import para copiar un proxecto ou árbore de directorios sen versionar dende o equipo local a un repositorio no servidor. A carpeta local queda asociada ao repositorio.
- Check out para copiar dun repositorio ao directorio local de traballo dun usuario. A carpeta local queda asociada ao repositorio.
- Commit para publicar no repositorio os cambios realizados no directorio de traballo local.
- Update para sincronizar ou actualizar o directorio de traballo local cos cambios acometidos no repositorio.

Import

É a operación a realizar cando o usuario ten un proxecto NetBeans creado localmente sen *versionar* (sen ter controladas as versións con Subversion) e quere *versionalo*, é dicir, subir unha copia ao servidor para controlar as versións dende ese servidor.

Para practicar con esta operación utilizarase un servidor Subversion na máquina virtual con IP 169.254.48.120 que ten creado o, coa estrutura de carpetas por defecto e ten os usuarios *angeles* e *manolo* autorizados para ler e escribir nese repositorio.

O usuario *angeles* ten un proxecto NetBeans Java denominado *sumarenteiros* en *C:\Users\Profesor\Documents\http\NetBeansProjects\sumarenteiros*.

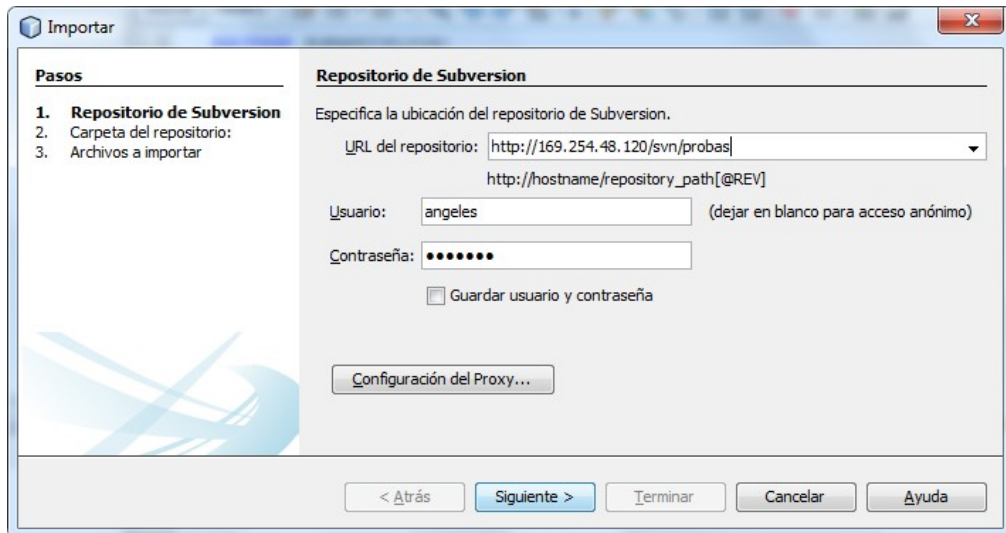


Para realizar import, débese seleccionar o proxecto *sumaenteiros* na ventá de proxectos e elixir entre:

Facer clic dereito, Versioning->Import into Subversion Repository...

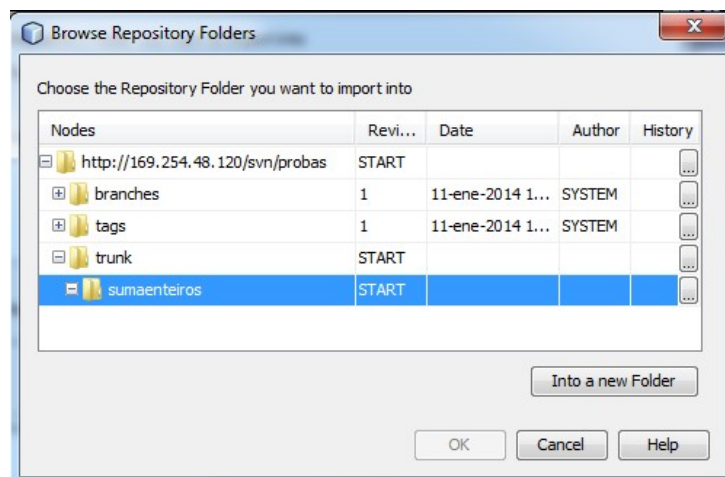
No menú principal: Team->Subversion->Importar al repositorio...

En ambos casos ábrese a ventá Import na que se debe teclear a url do repositorio e o nome e contrasinal do usuario cos permisos necesarios.

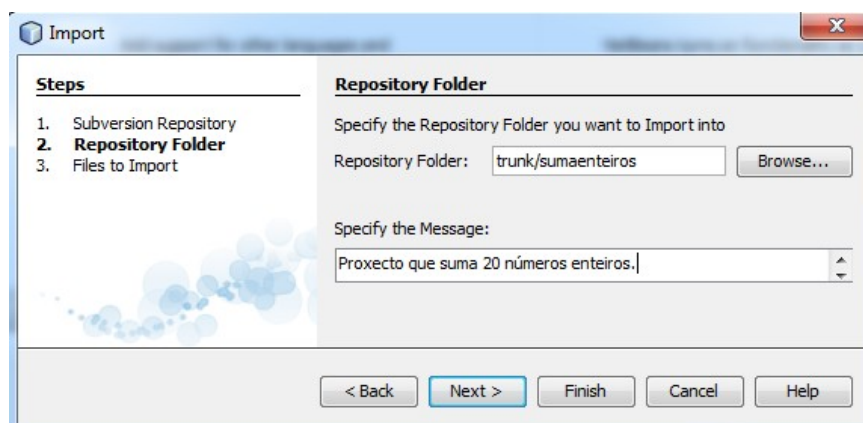


Na seguinte ventá hai que indicar:

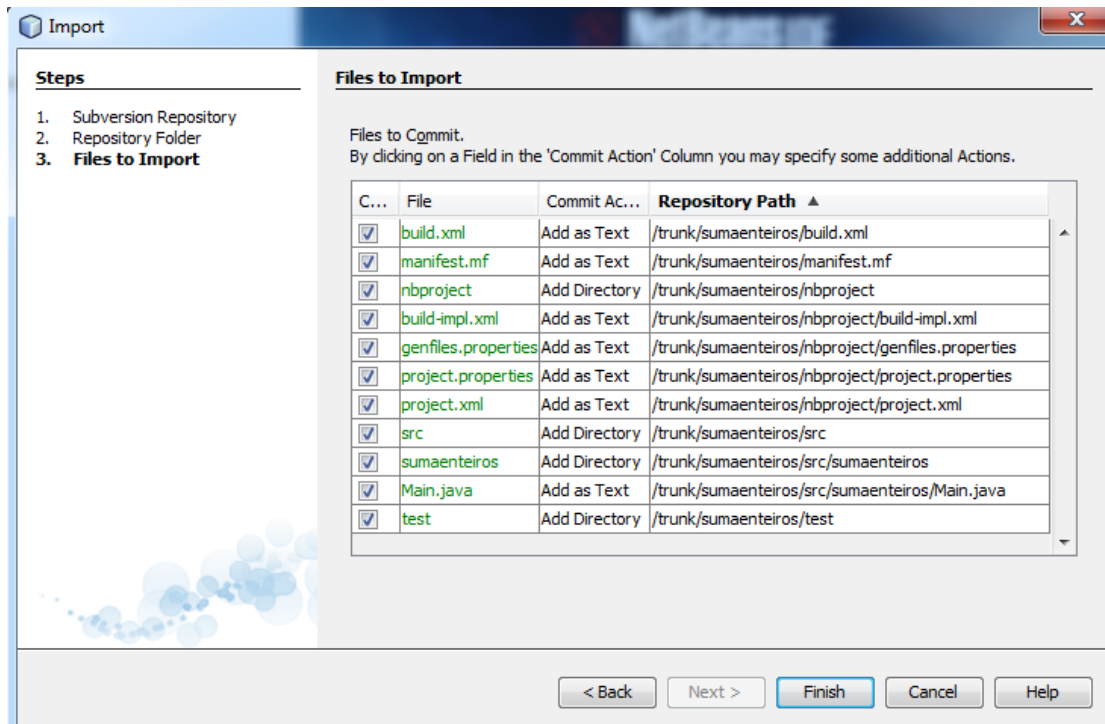
- A carpeta na que se quere gardar o proxecto dentro do repositorio. Suxire unha carpeta co mesmo nome que a carpeta local e paralela ás carpetas do repositorio. Pódese premer en *Browse* para elixir comodamente a carpeta desexada. Nesta práctica elixirase por exemplo a carpeta trunk e pulsando *A una carpeta nueva* creárase unha carpeta co mesmo nome que a carpeta local *sumaenteiros*.



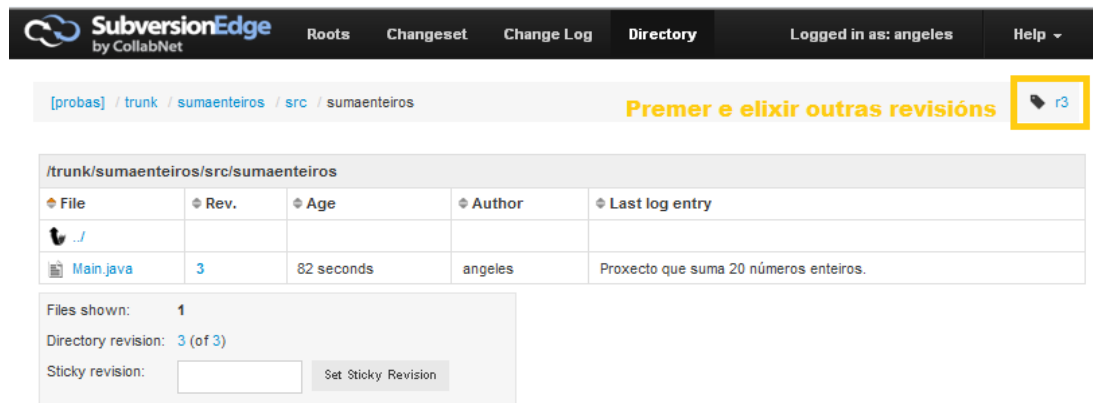
- A descrición do proxecto.



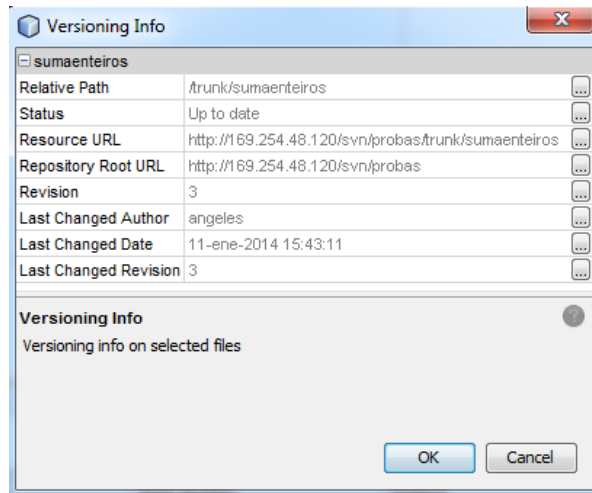
- Opcionalmente pódese premer en *Siguiente*> e elixir ter unha vista previa do resultado final no repositorio.



- Faise clic en *Terminar* para iniciar a importación e pode comprobarse no servidor se está subido o proxecto como revisión 3. Pode verse que a revisión 1 foi creada automaticamente cando se creou a estrutura inicial de carpetas do repositorio e a revisión 2 foi creada automaticamente cando se creou a carpeta *sumaateiros* dentro da carpeta *trunk*.



Pode verse a información da versión iluminando o proxecto, clic dereito *Subversion->Versioning Info...*



Checkout

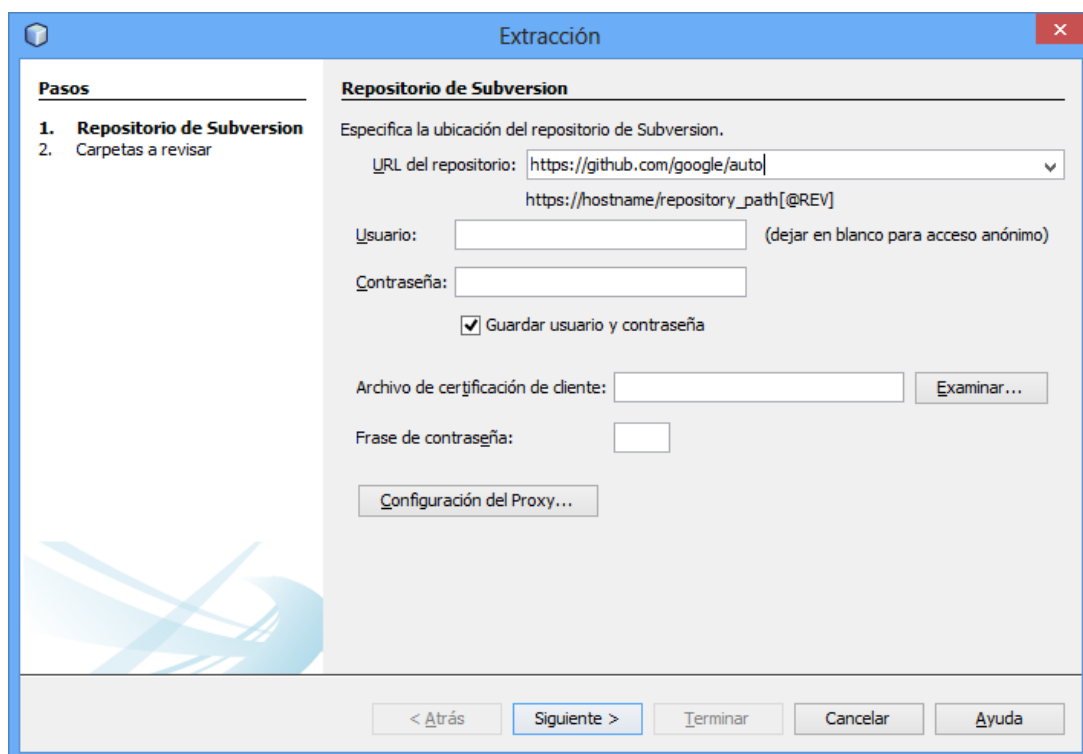
Esta operación fai unha copia local dun proxecto ou conxunto de arquivos que están nun repositorio Subversion.

A primeira práctica con checkout que se vai realizar consiste en extraer o proxecto Java google-api-java-client do repositorio de github (<https://github.com/google/auto>)

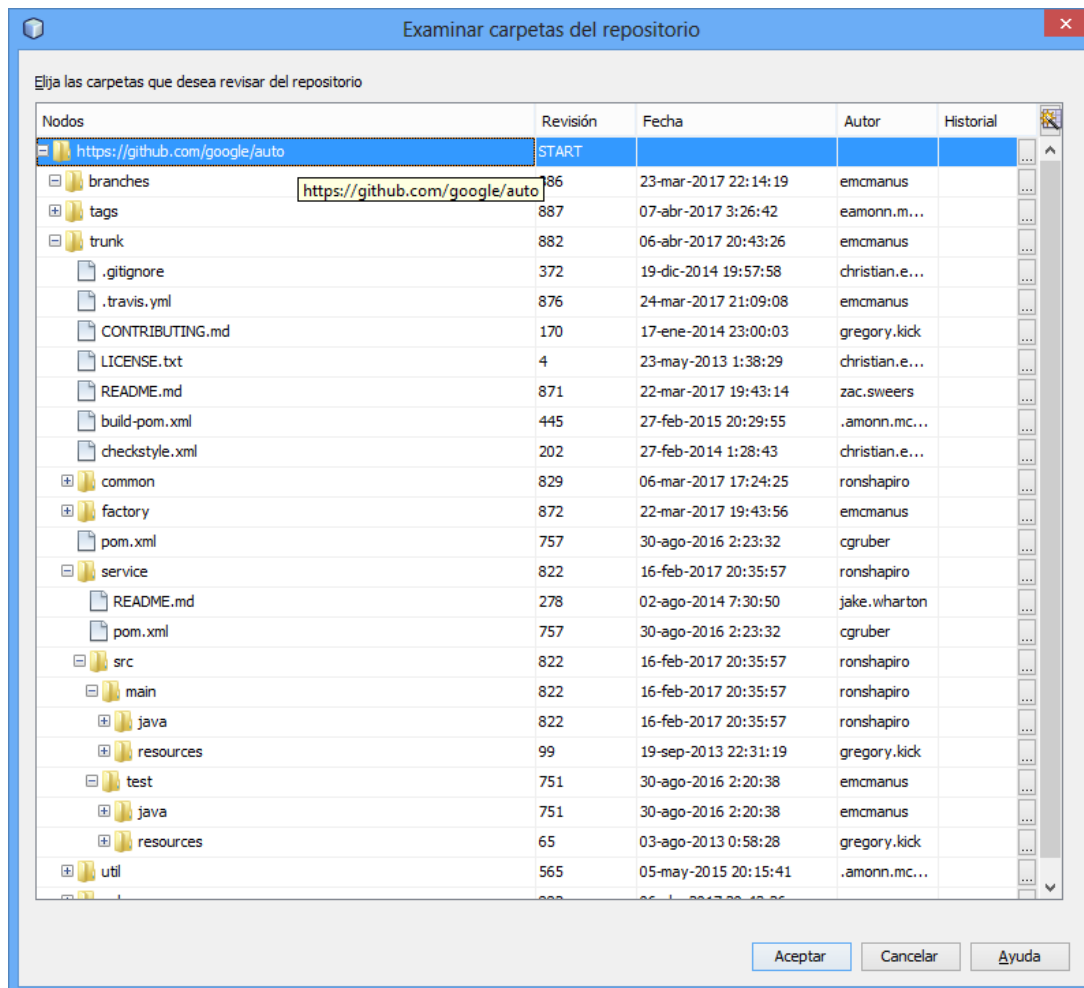
Para descargalo e asociar a carpeta local co repositorio hai que elixir:

Team->Subversion->Checkout

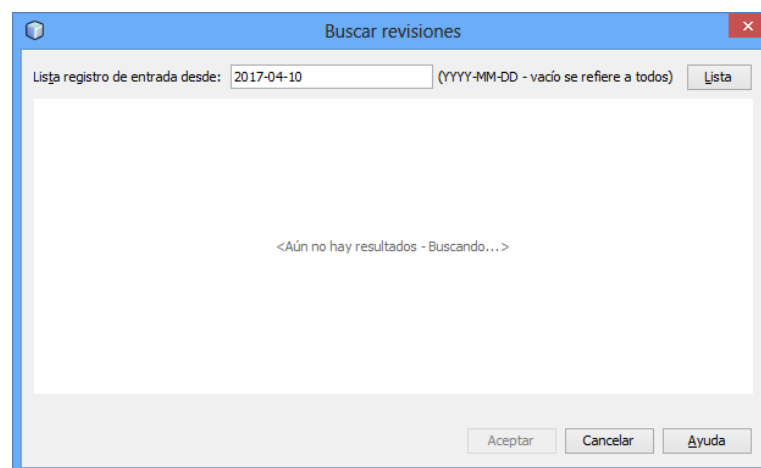
e na ventá *Checkout* teclear a URL do repositorio Subversion e o usuario autorizado coa súa contrasinal se o acceso non é anónimo.



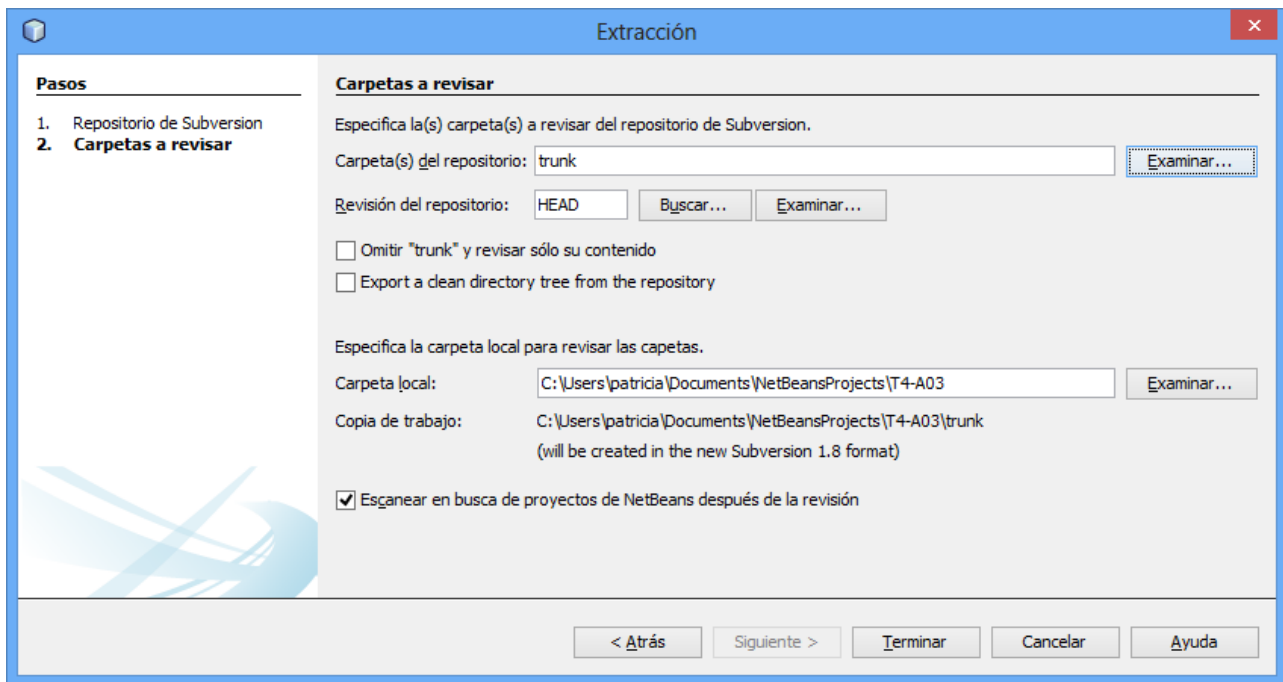
O seguinte paso é detallar as carpetas do repositorio que se queren extraer se non se quere extraer todo. O exame do repositorio pode servir para seleccionalas de maneira cómoda e segura.



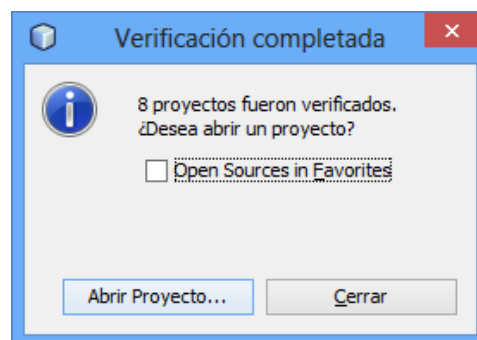
Pódese detallar a revisión desexada se é que se desexa unha diferente da versión head



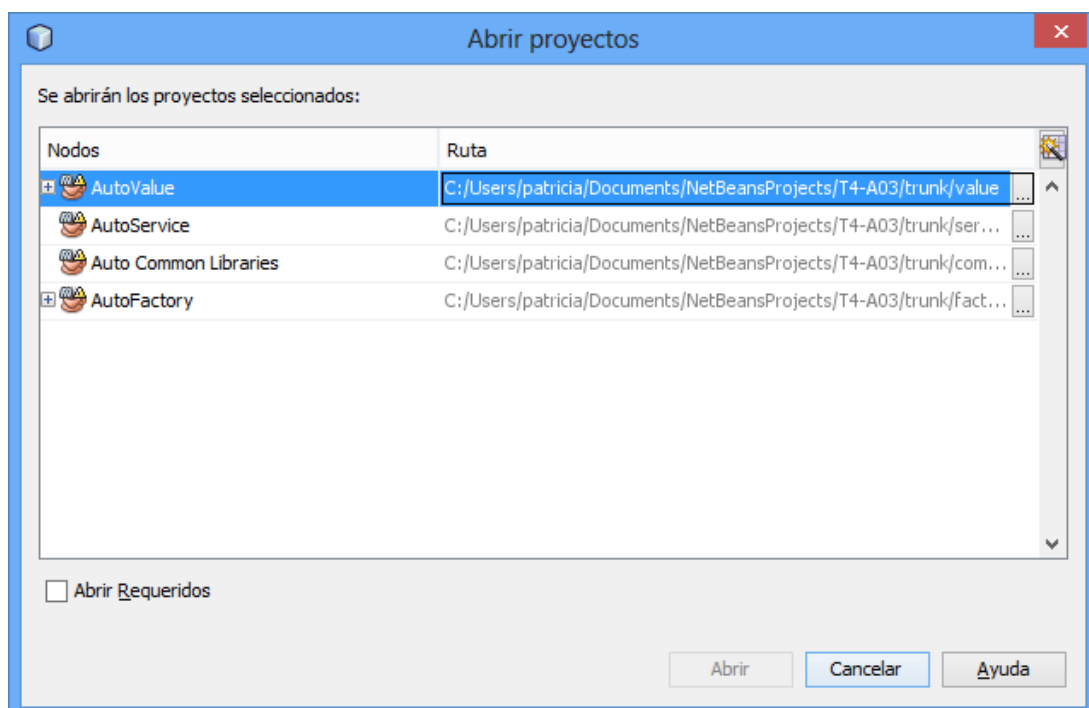
e a carpeta local onde se gravará o código extraído.



Cando termina a extracción, e se o código extraído é un proxecto NetBeans, pódese abrir o proxecto;



Se prememos en Abrir proyecto

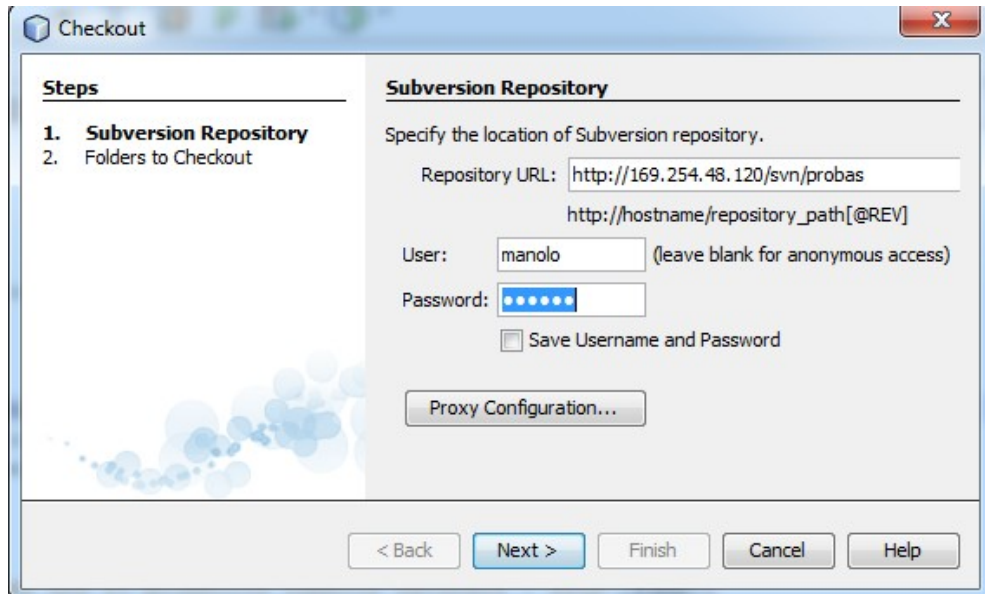


se non é un proxecto NetBeans, pode indicarse que se quere formar un proxecto, seleccionar a categoría de proxecto adecuado e indicar que se quere abrir. En calquera caso, o proxecto queda aberto.

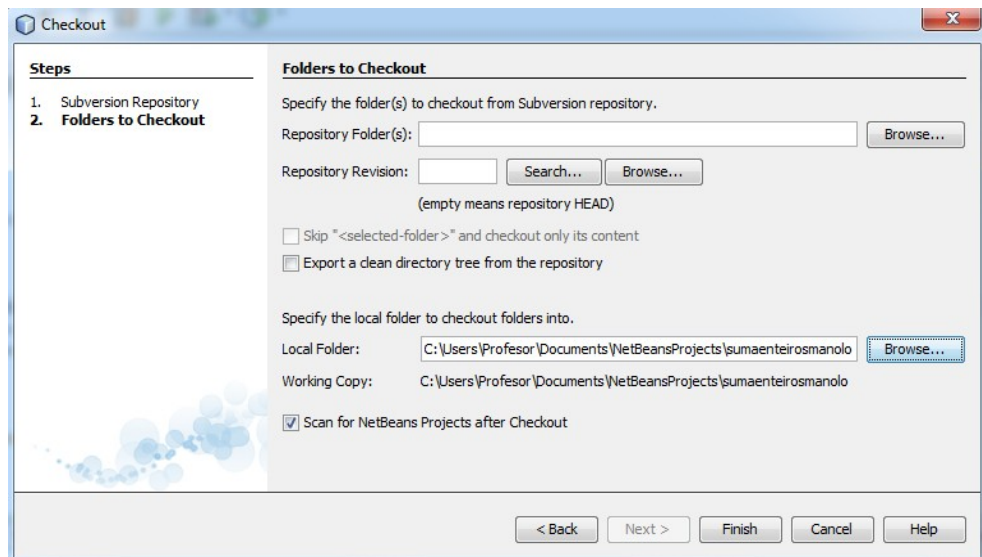
A segunda práctica con *checkout* consiste en que o usuario *manolo* extraia o proxecto Java sumaenteiros do repositorio probas do servidor Subversion da máquina virtual con IP 169.254.48.120 co que se traballou na primeira práctica deste documento. Para iso elíxese:

Team->Subversion->Checkout

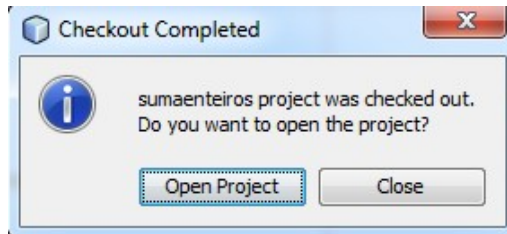
e na ventá Checkout indícase a URL do repositorio e o usuario *manolo*.



Vaise copiar todo o repositorio no directorio de traballo (*sumaenteirosmanolo*).



NetBeans detecta que é un proxecto NetBeans e pregunta se se quere abrir.



Na ventá Output poden verse as operacións que se van realizando.

Commit

É a operación que permite publicar no servidor os cambios realizados no directorio local de traballo.

Para practicar con esta operación o usuario *manolo* realiza modificacións no código de Main.java. Aparecen símbolos á esquerda das liñas de código indicando se a liña foi engadida (cor verde) ou modificada (cor azul).

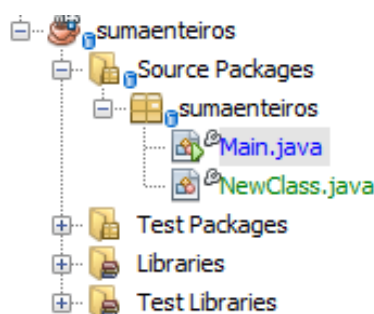
```

public class Main {






    public static void main(String[] args) {
        int suma=0;      /* sumador dos números enteiros */
        int producto=1;
        for (int i=1;i<=20;i++)
        {
            suma=suma-i;
        }
        System.out.printf("Suma dos 20 primeiros números naturais = %d\n", suma);
    }
}

```



O usuario *manolo* pode crear un arquivo novo. Na ventá de proxectos aparece o novo arquivo en cor verde e a carpeta cos arquivos fontes aparece cunha icona en azul indicando que os cambios non están publicados no servidor Subversion e por tanto o proxecto tamén aparece coa mesma icona.



O significado das cores utilizadas é o seguinte:

Color Coding	Description
 Main.java	Green. Indicates that the file is a new local file that does not yet exist in the repository.
 Main.java	Blue. Indicates that the file has been modified locally.
 Main.java	Red. Indicates that the file contains conflicts. You must employ the Resolve Conflicts command (Subversion > Resolve Conflicts) for such files.
 Main.java	Grey. Indicates that the file is ignored by Subversion and will not be included when calling versioning commands. In the Versioning window , grey text signifies deleted files.
 Main.java	Strike-through. Indicates that the file is excluded when calling the Commit command. All other Subversion commands, however, work as usual. Note that files displayed in the strike-through style only appear in the Versioning window and Commit dialog. They will not appear in Diff panes, nor will their parent folders (or packages) display badges if they are modified.

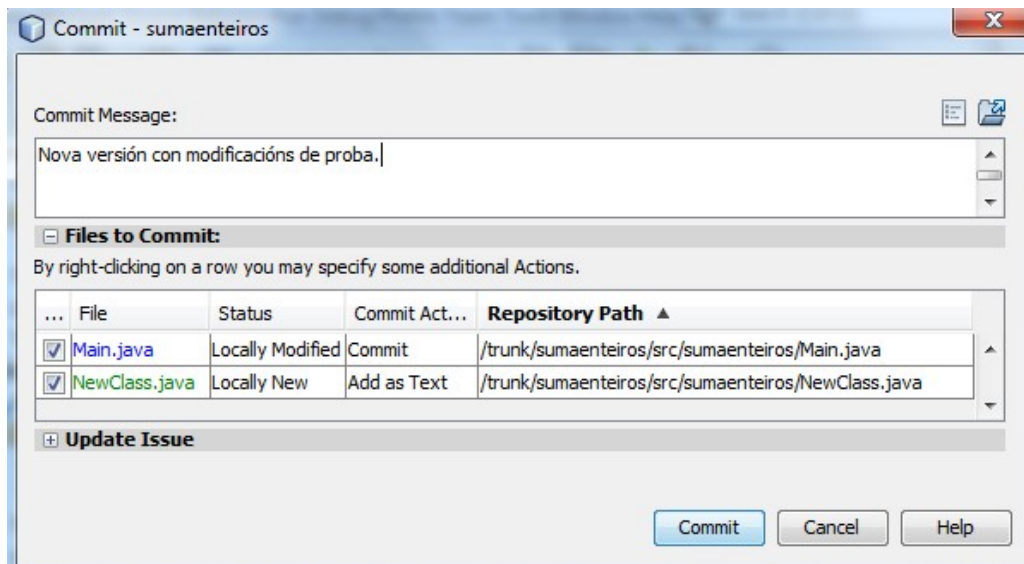
O significado dos símbolos que aparecen ao carón de arquivos, carpetas ou proxecto é o seguinte:

Badge	Description
 myapplication	Locally Modified Badge. A blue badge on a folder or package node marks folders or packages that contain locally modified or new files. In the case of packages, this badge applies only to the package itself and not its subpackages. For folders, the badge indicates local modifications in that folder or any of its subfolders.
 myapplication	Conflict Badge. A red badge on a folder or package node marks folders or packages that contain files for which the repository copy of the file contains changes which conflict with the local version. In case of packages, this badge applies only to the package itself and not its subpackages. For folders, the badge indicates local modifications in that folder or any of its subfolders.

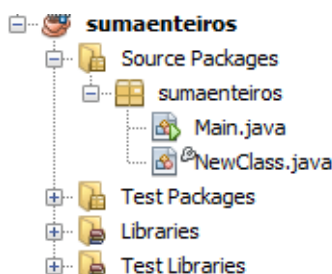
Antes de publicar os cambios no servidor, *manolo* podería ver se houbo cambios no repositorio dende que el fixo o último Checkout e para iso ilumina o proxecto *versionado* na ventá de proxectos, fai clic dereito -> *Subversion*-> *Mostrar cambios*. Na ventá Subversion aparecen os arquivos que están modificados en cor azul e os novos en cor verde.

Subversion - sumaenteiros (21 seconds ago)		
File	Status ▲	Repository Location
NewClass.java	Locally New	/trunk/sumaenteiros/src/sumaenteiros/NewClass.java
sumaenteiros	Locally Modified	/trunk/sumaenteiros

O usuario *manolo* ten que facer *commit* para actualizar o repositorio cos cambios locais e para iso ilumina o proxecto *sumaenteiros*, fai clic dereito -> *Subversion* -> *Confirmar...* e teclea a descrición desta nova versión.



Na ventá de proxectos desaparecen as marcas de arquivos e carpetas pendentes de publicar.



Update

É a operación que permite actualizar o directorio de traballo coas modificacións do repositorio.

Para practicar con esta operación, o usuario *angeles* deberá de iluminar o seu proxecto, facer clic dereito -> *Subversion*-> *Update*. Pode observarse que os arquivos locais teñen as modificacións feitas polo usuario manolo.

Diff

A operación *commit* pode provocar conflitos por diferencias incompatibles entre o contido dalgún arquivo local e a versión almacenada no repositorio. Neste caso Subversion informa inmediatamente ao usuario da existencia dun conflito, e propón diversas accions a realizar en ese momento. As opcións posibles son: descartar os cambios locais, descartar os cambios do repositorio e fusionar as dúas versións nunha nova e facer *commit* da nova versión.

A recomendación para evitar conflitos é actualizar o directorio de traballo xusto antes de poñerse a traballar con ela, subir as modificacións canto antes, consultar as diferencias entre o contido do directorio de traballo e o repositorio e facer *commit*.

Para practicar este apartado, provocarase que o usuario *angeles* e *manolo* partan da mesma versión do repositorio no seu directorio de traballo, que *angeles* faga *commit* con algunhas modificacións, que *manolo* non actualice o directorio de traballo e faga outras modificacións e despois cando tente facer *commit* xurdirá o conflito. Para iso faranse as seguintes operacións:

O usuario *angeles* borra o arquivo *NewClass.java* e modifica unha liña de *Main.java*.

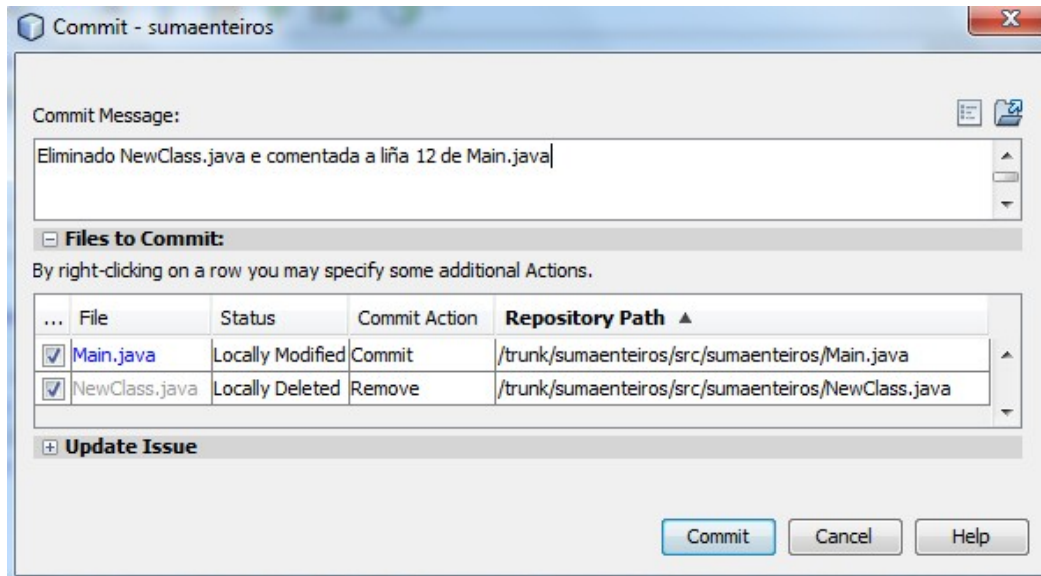
```

public class Main {

    public static void main(String[] args) {
        int suma=0;        /* sumador dos números enteiros */
        /*int producto=1;*/
        for (int i=1;i<=20;i++)
        {
            suma=suma-i;
        }
        System.out.printf("Suma dos 20 primeiros números naturais = %d\n",suma);
    }
}

```

O usuario *angeles* fai *commit*.



O usuario *manolo* non baixa a última versión do repositorio e modifica Main.java borrando unha liña, modificando outra e aumentando outra.

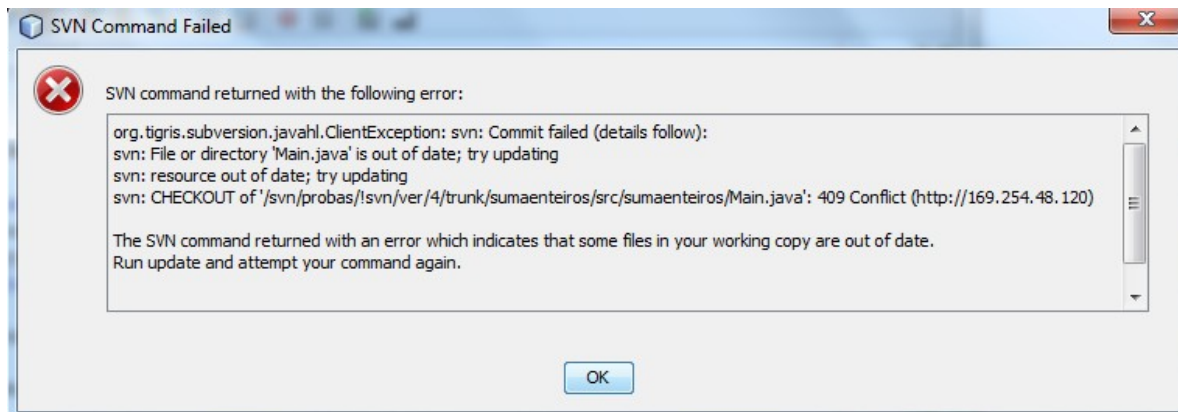
```

public class Main {

    public static void main(String[] args) {
        int suma=0;        /* sumador dos números enteiros */
        for (int i=1;i<=40;i++)
        {
            suma=suma-i;
        }
        /* aumenta liña */
        System.out.printf("Suma dos 20 primeiros números naturais = %d\n",suma);
    }
}

```

O usuario *manolo* fai *commit* e SVN avisa do conflito co arquivo Main.java e SVN suxire facer *update*.



A forma máis rápida de ver en detalle as diferencias entre dúas versións de arquivos (loais, do repositorio, entre directorio de traballo e repositorio) é seleccionar o arquivo por exemplo na ventá do proxecto, facer clic dereito -> *Subversion*->*Diff*->Diferencia e aparecería directamente a pestana das diferencias.

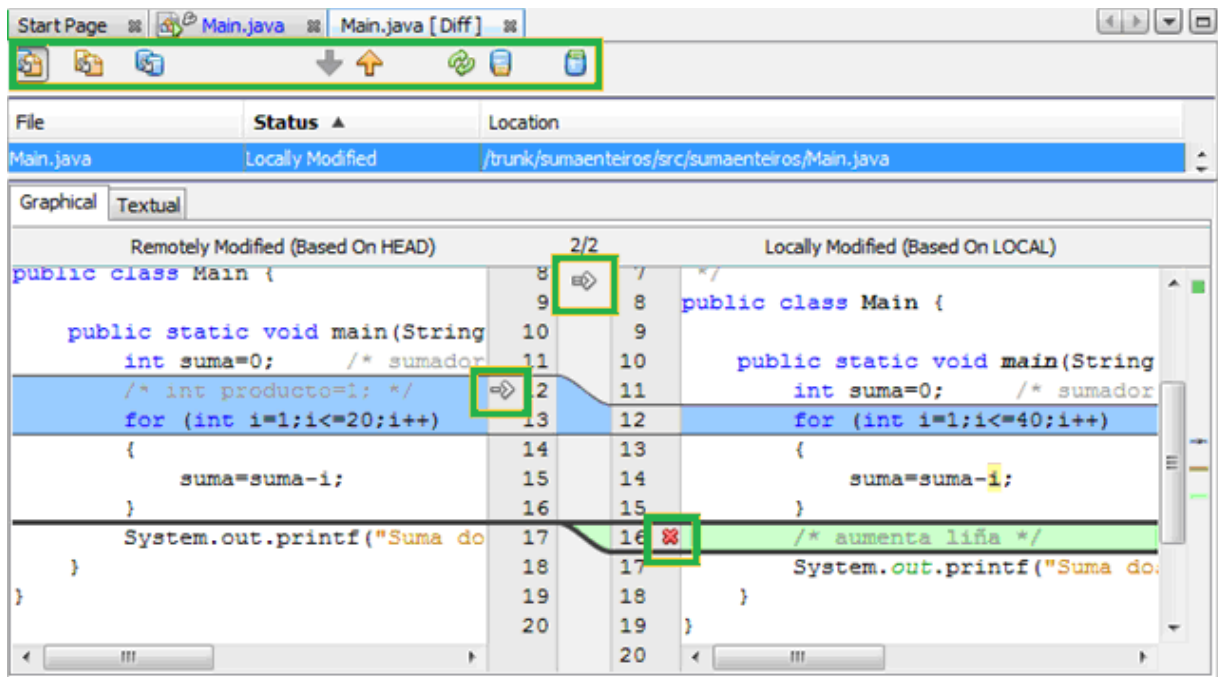
A pestana das diferencias ten unhas iconas na parte superior esquerda que permiten:

- Ver as diferencias entre o arquivo local e o do repositorio.
- Ver as diferencias entre as dúas últimas versións en local.
- Ver as diferencias entre as dúas últimas versións no repositorio.
- Moverse cara abaixo ou cara arriba entre as diferencias.
- Actualizar as diferencias.
- Facer Update.
- Facer Commit.


Na pestana das diferencias aparecen unhas frechas ao carón dos números de liñas con diferencias que permiten facer cambios directamente:

Replace (↔)
Move All (⇄)
Remove (✖)

Na pestana de diferencias, pódense facer cambios manuais no panel da dereita (en local) que serán salvados automaticamente.

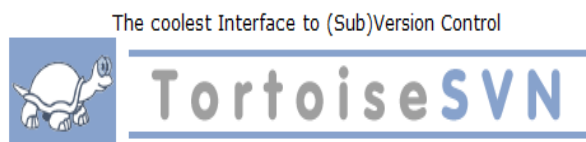


Unha solución ao problema do usuario *manolo* pode ser facer unha copia do arquivo local con outro nome para preservar o seu traballo, facer *update* para ter no directorio de traballo a última versión do repositorio, fusionar manualmente coa copia e facer *commit* para publicar as súas modificacións.

 Tarefa 2. Exercicios para practicar con Subversion en NetBeans.

TortoiseSVN

Introdución



É un cliente de Subversión, implementado como extensión do shell de Windows.

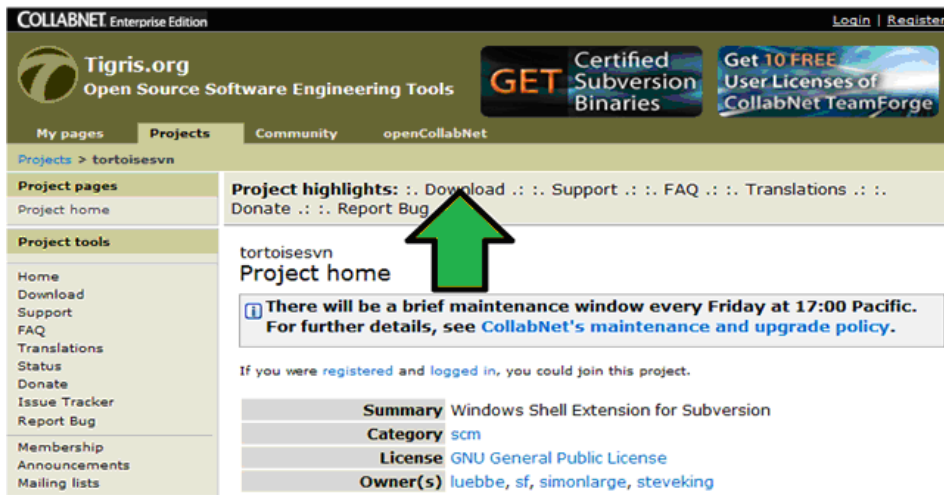
Con TortoiseSVN é realmente fácil controlar as versións e modificacións de arquivos ou de código fonte do software en entorno Windows.

Posto que non é unha integración para un IDE específico pódese usar con calquera ferramenta de desenvolvemento.

TortoiseSVN é software libre.

Instalación de TortoiseSVN

Descárgase dende a url <http://tortoisesvn.tigris.org>



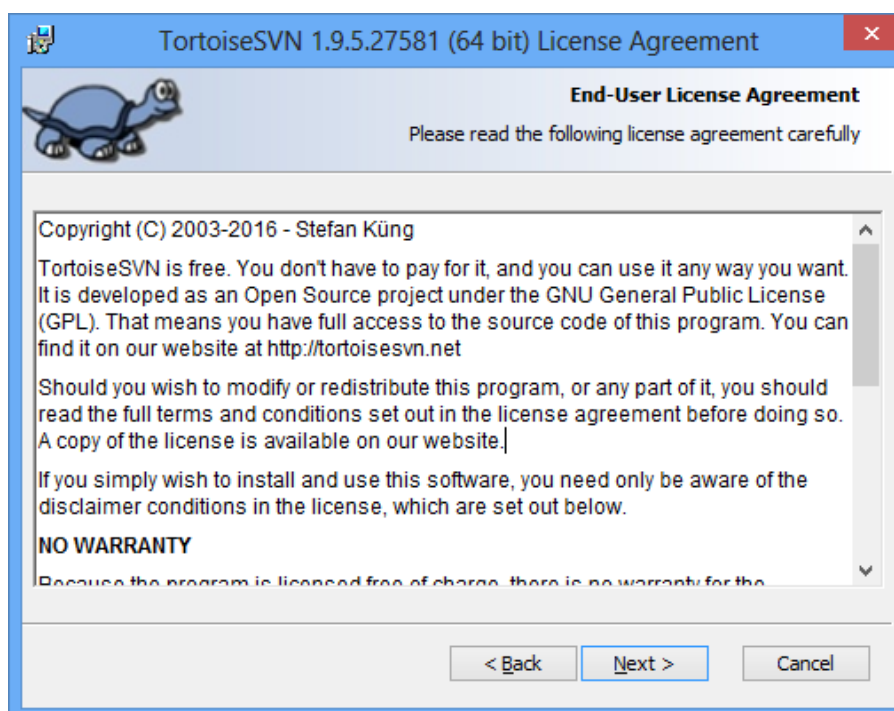
Neste caso a instalación vaise facer sobre unha máquina virtual con Windows 7 de 32 bits. Descárgase a versión para 32 bits.



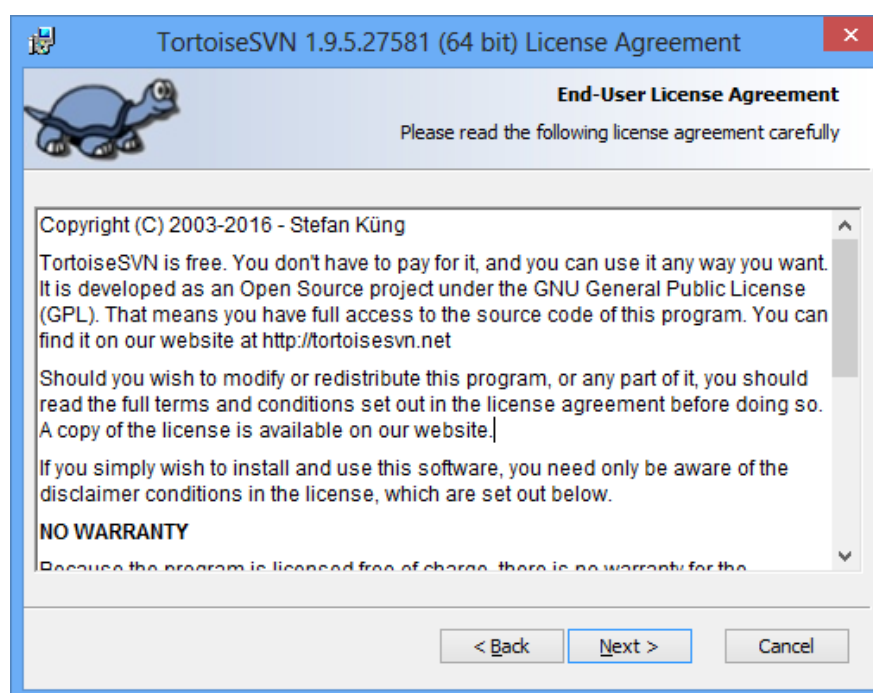
Comeza a instalación.



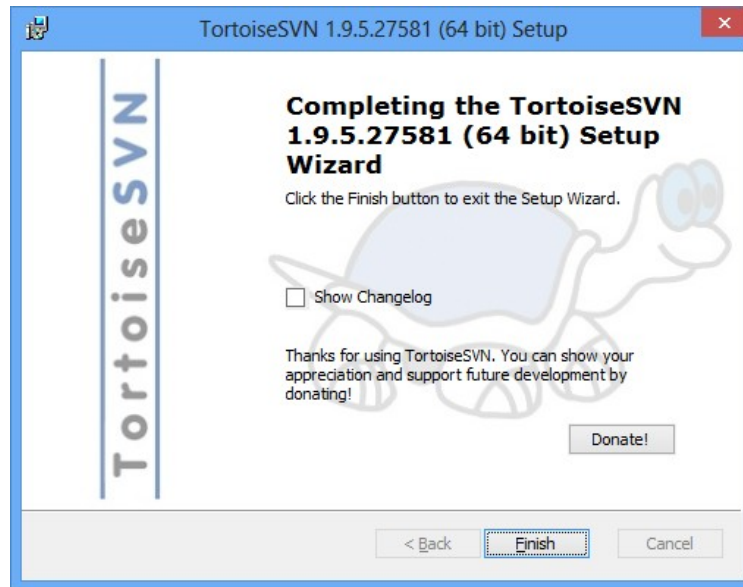
Lese e acéptase a licenza.



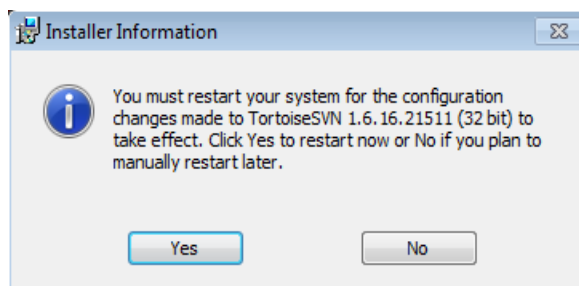
Déixanse os componentes que instala por defecto e a carpeta de instalación por defecto.




Remata a instalación.



É necesario reiniciar o equipo.

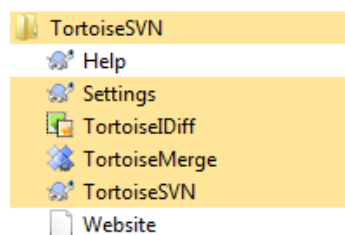


 Tarefa 3: Instalación de TortoiseSVN nunha máquina virtual Windows 7 de 32 bits

Utilización de TortoiseSVN

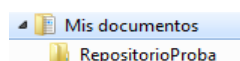
O cliente Tortoise queda integrado no explorador de arquivos de Windows e, por tanto, pódese operar desde os menús contextuais facendo clic co botón dereito do rato.

Poden realizarse operacións con Tortoise desde o menú Inicio.



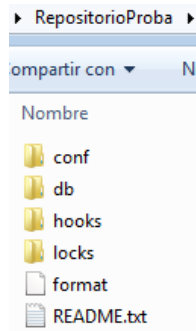
Utilización de Tortoise en local. [Create repository here.](#)

Crear a carpeta RepositorioProba



Para crear nela un repositorio, débese de iluminar, clic dereito, e elixir *TortoiseSVN*->*Create repository here*.

Tortoise crea na carpeta a estrutura necesaria para administrar o repositorio.

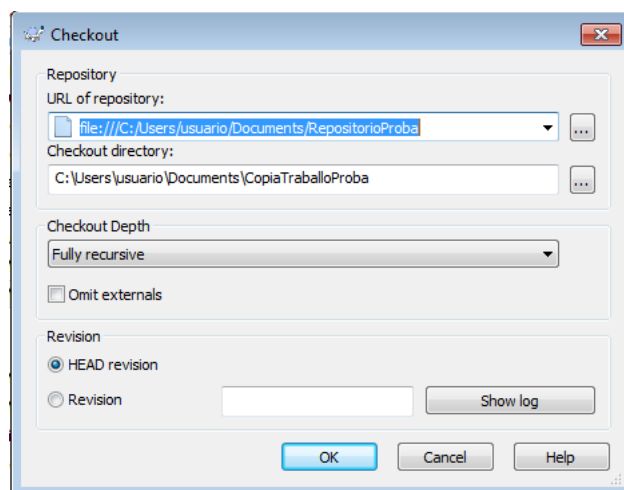


Checkout

Crear a carpeta *CopiaTraballoProba* para utilizar como directorio de traballo do repositorio Proba.

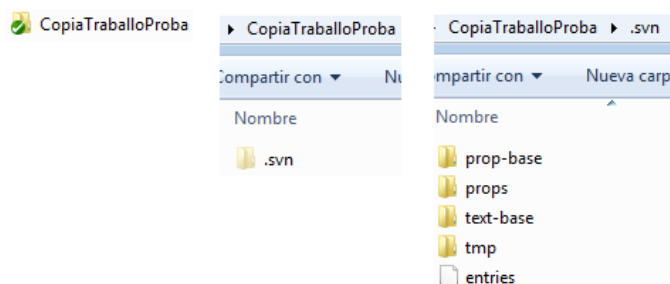
Para sincronizar esta carpeta co repositorio Proba, débese iluminar a carpeta, clic dereito e elixir *SVN Checkout*....

Utilízase o protocolo file e faise checkout total da última revisión que neste caso está baleira.

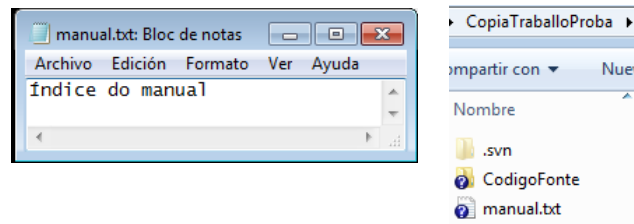


Carpeta de traballo

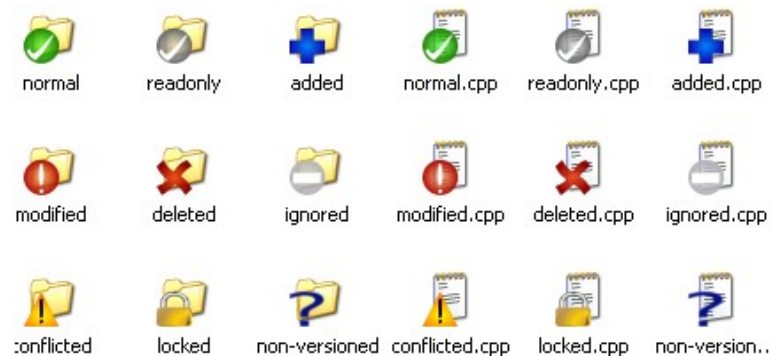
Tortoise crea na carpeta de traballo a carpeta oculta *.svn* para administrar a copia de traballo local e a relación co repositorio.



Ao crear un arquivo ou unha carpeta dentro do directorio de traballo aparecerá como non versionado.



Posibles identificacións de carpetas e arquivos na carpeta de traballo:



Add

Para versionar a carpeta ou o arquivo deben de engadirse para que formen parte do repositorio cando se publiquen os cambios; para iso debe de iluminarse a carpeta ou o arquivo, clic dereito e elixir *TortoiseSVN* -> *Add*.

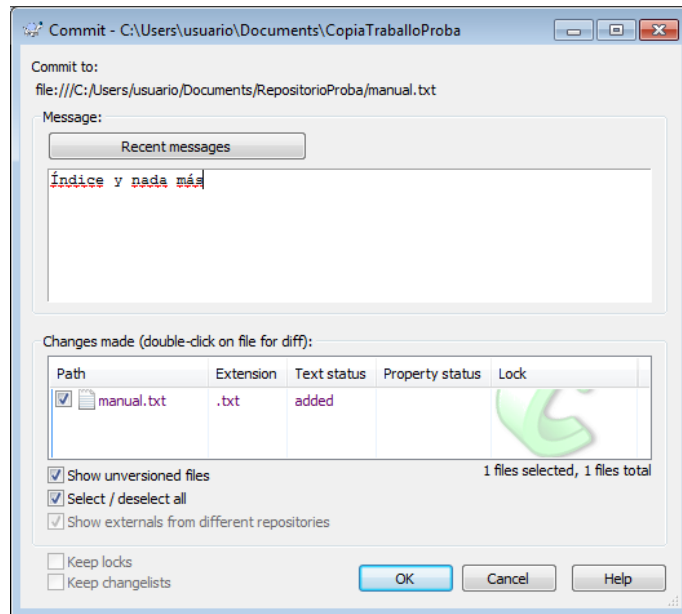
A carpeta e o arquivo engadidos aparecen adecuadamente identificados.

O directorio de traballo aparece identificado como modificado con relación á última versión baixada do repositorio.



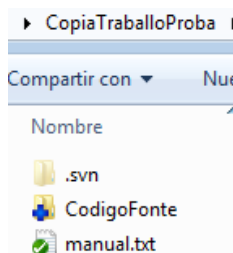
Commit

Para publicar o arquivo ou a carpeta dende a copia de traballo ao repositorio débese Iluminar, clic dereito e elixir *SVN Commit*.

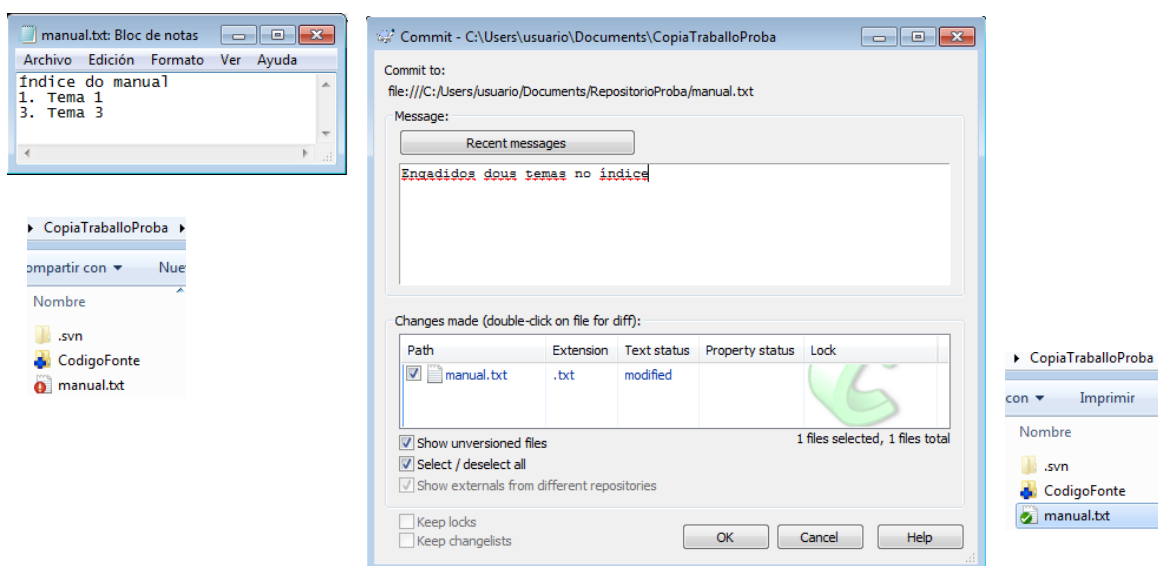


Débese de teclear a mensaxe asociada á revisión.

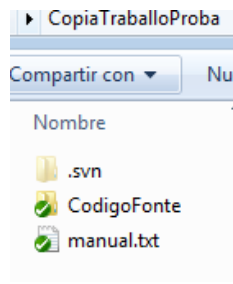
Isto crea unha revisión (neste caso a revisión 1), é dicir, unha instantánea dese momento.



Pódense facer modificacións sobre o arquivo de traballo e publicar esas modificacións creando a versión 2.

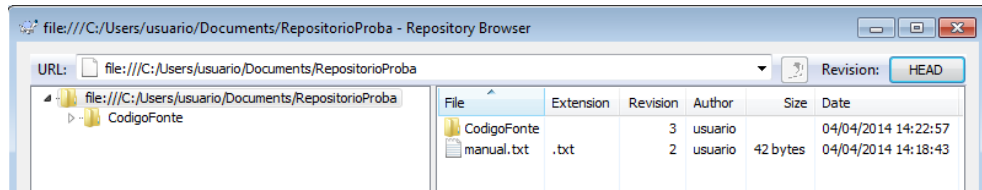


Pódese publicar a carpeta de arquivos fonte e crear a revisión 3.



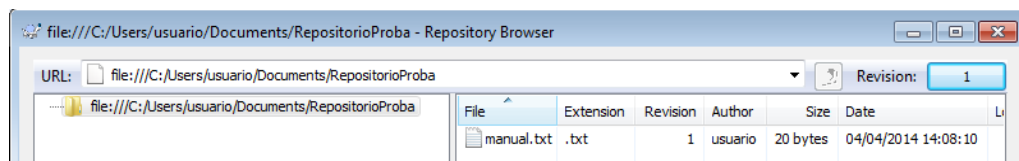
Repo-browser

Para ver as revisións existentes no repositorio débese de iluminar o directorio de traballo e elixir *TortoiseSVN* -> *Repo-browser*.



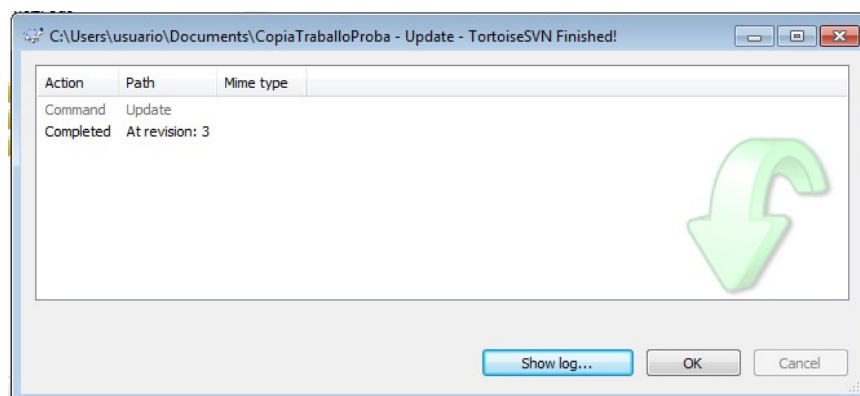
Pode verse que a revisión HEAD (a última) do repositorio é a revisión 3 que afecta á carpeta CodigoFonte, e a revisión 2 afecta ao arquivo manual.txt.

Para ter información sobre a revisión 1, debe de pincharse no botón Revisión:HEAD e cambiar a revisión pola 1.



Update

Para actualizar o directorio de traballo coa última revisión do repositorio, débese de iluminar o directorio de traballo, clic dereito, elixir *SVN Update*.



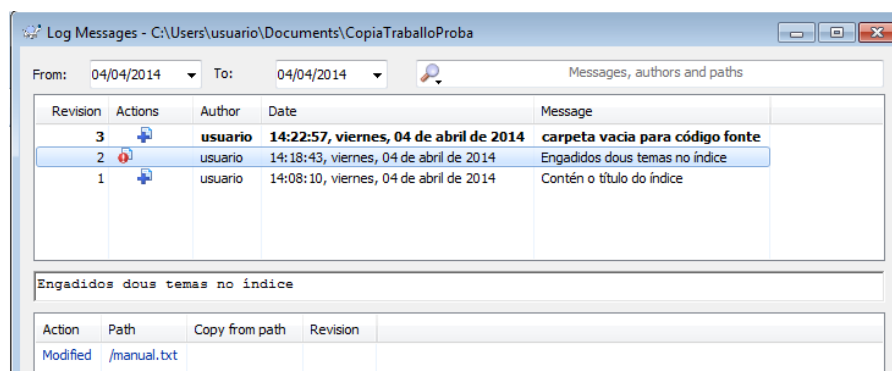
O botón *Show log...* permite ter máis información sobre as distintas revisións.

Show log

Permite ver información detallada das revisións do elemento iluminado.

Pode accederse dende a ventá de *Update* ou dende o menú *TortoiseSVN* -> *Show log*.

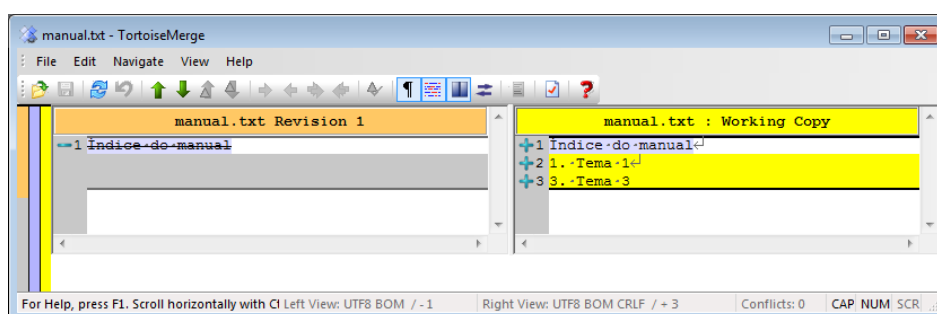
Por exemplo para obter información sobre todo o repositorio, ilumínase a carpeta do directorio de traballo, clic dereito, *TortoiseSVN -> Show log*



Diff

Para ver as diferenzas entre un elemento do directorio de traballo e a revisión anterior, débese de iluminar o elemento, clic dereito e elixir

TortoiseSVN -> Diff with previous version...

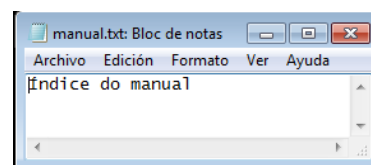
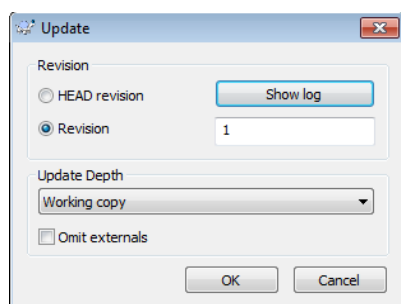


Update to revision

Para actualizar o directorio de traballo con calquera revisión, débese de iluminar o directorio de traballo, clic dereito e elixir

TortoiseSVN -> Update to revision...

Se a revisión elixida é a revisión 1, pode editarse o arquivo e comprobar que ten o contido inicial.



Tarefa 4: Utilización de Tortoise en local. Create repository here.

Utilización de Tortoise. Cliente e servidor na mesma máquina.

Neste caso utilizarase o repositorio *manuais* do servidor Subversion instalado nunha máquina virtual, e simularase a utilización do mesmo por dous usuarios autorizados: *manolo* e *angeles* que teñen no disco da máquina virtual dúas carpetas sincronizadas con ese repositorio. Para iso:

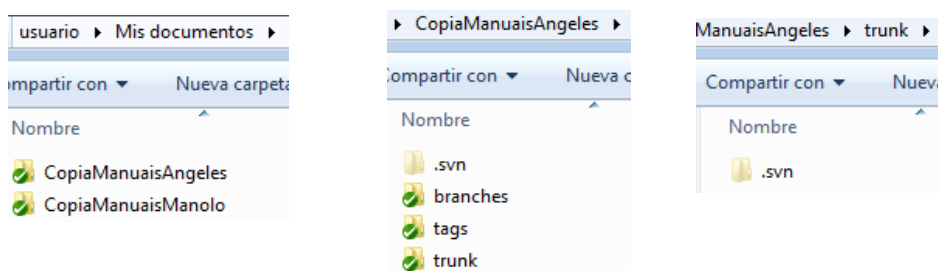
- No servidor Subversion hai que crear o repositorio *manuais* coa estrutura de carpetas por defecto, os usuarios *manolo* e *angeles* con roles de usuario, e as regras de acceso que permiten ter un grupo formado polos usuarios *manolo* e *angeles* con permisos rw sobre o repositorio.
- No disco local hai que crear dúas carpetas : *CopiaManuaisAngeles* e *CopiaManuaisManolo*.

```
[groups]
grupomanuais = manolo, angeles

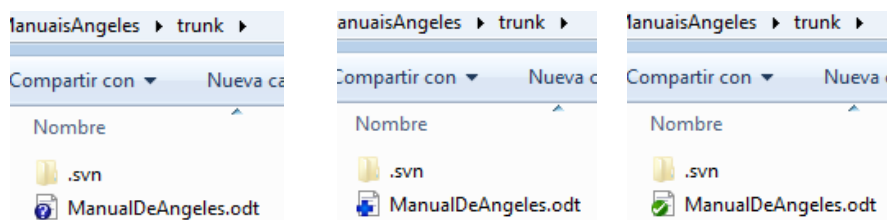
[manuais:/]
@grupomanuais = rw
```

CopiaManuaisAngeles
CopiaManuaisManolo

As carpetas de traballo terán a carpeta oculta *.svn* e a estrutura de carpetas por defecto copiada dende o repositorio.



Cada un dos usuarios crea na carpeta de traballo trunk un arquivo diferente, engádeo a Tortoise e o publica no repositorio. Obsérvase que a revisión 1 foi creada ao crear a estrutura de carpetas no repositorio.

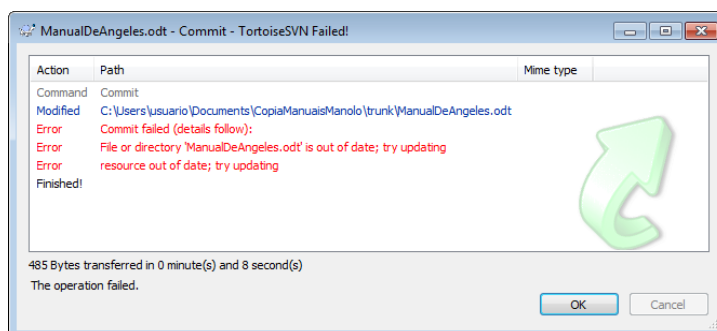


Conflicto

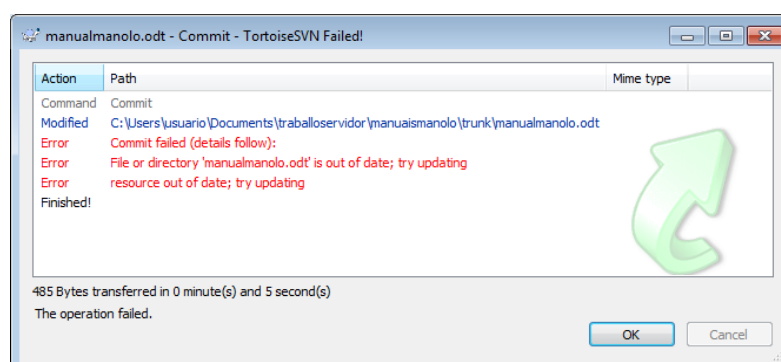
Para provocar un conflito faremos que:

- O usuario manolo actualice (Update) a súa copia de traballo, é dicir, terá agora os arquivos ManualdeAngeles.txt e ManualdeManolo.txt.
- O usuario angeles faga modificacións sobre o seu manual e as publique.
- O usuario manolo modifique o manual de angeles da súa copia de traballo (sen ter a precaución de comprobar que non está traballando sobre a última versión) e intente publicala.

Subversion avisará dun erro xa que a copia que está intentando publicar está baseada nunha versión que non é a última no repositorio.



O usuario *manolo* tenta publicar o seu manual pero detéctase un conflito que debe ser resolto manualmente e queda sen facer a publicación.



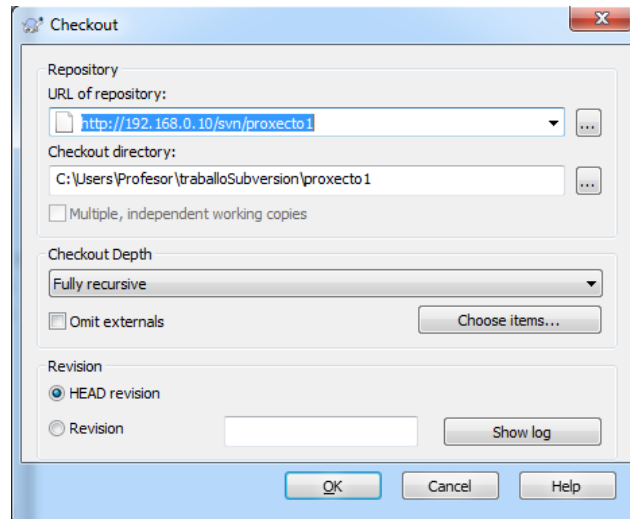
Tarefa 5: Utilización de Tortoise. Cliente e servidor na mesma máquina.

Utilización de Tortoise. Cliente e servidor en diferentes máquinas virtuais e mesmo ordenador

Este caso diferénciase do anterior basicamente en que o cliente comunícase co servidor coa URL <http://nomehost/svn/rutarepositorio>

Se non existe servidor DNS que asocie a dirección IP da máquina virtual co *nomehost*, haberá que indicar a dirección IP.

Para facer checkout, selecciónase o directorio de traballo, clic dereito, *checkout* e tecléase a URL do repositorio e os demais detalles do *checkout*.



1.3 Tarefas

As tarefas propostas son as seguintes.

- **Tarefa 1.** *Instalación de servidor Subversion CollabNet Subversion Edge*
- **Tarefa 2.** *Exercicios para practicar con Subversion en NetBeans.*
- **Tarefa 3.** *Instalación de TortoiseSVN nunha máquina virtual Windows 7 de 32 bits.*
- **Tarefa 4.** *Utilización de Tortoise en local. Create repository here.*
- **Tarefa 5.** *Utilización de Tortoise. Cliente e servidor na mesma máquina.*

1.3.1 Tarefa 1. Instalación de servidor Subversion CollabNet Subversion Edge

1.3.2 Tarefa 2. Exercicios para practicar con Subversion en NetBeans.

- Utilizar un repositorio de internet.
 - A url é
<https://github.com/google/auto>
 - Executar o proxecto anterior e comprobar que funciona.
- Crear repositorio na máquina virtual.
 - Crear un novo proxecto no administrador do servidor subversión denominado *suma*
 - Crear dous usuarios con permisos de lectura e escritura no acceso o proxecto *suma*.
- Utilizar un repositorio Subversión creado na máquina virtual dende a máquina real:
 - Crear un proxecto no Netbeans denominado *suma*. No menú *Control de versiones/Subversion/Importar al deposito* importa o proxecto netbeans *suma* no proxecto *suma* do repositorio.
 - Modifica o programa para que pida dous enteiros do teclado e mostre a súa suma.
Url svn://ip_maquina_virtual/svn/suma
 - Actualiza o proxecto no servidor subversion facendo *Commit*
 - Crea un novo proxecto no administrador do servidor subversión denominado *proxectoweb*

- Fai un *checkout* deste proxecto (en principio baleiro) na carpeta *proxectoweb1* no escritorio. Para facer o checkout terás que poñer na caixa de texto *svn://ip_maquina_virtual/svn/proxectoweb*
- Modifica o proxecto que está baleiro engadindo unha web *index.html* que só mostre un texto.
- Fai *commit* do *proxectoweb*.
- Fai *checkout* de novo o *proxectoweb* pero dentro doutra carpeta *proxectoweb2* no escritorio.
- Comproba que o arquivo *index.html* está correcto.
- Modifica *index.html* tanto en *proxectoweb1* como en *proxectoweb2* pero en distinto número de liña.
- Fai *commit* de ambos proxectos.
- Fai *update* de ambos proxectos. Comproba que mesturou ben os cambios.
- Modifica ambos proxectos pero agora na mesma liña.
- Fai *commit* de ambos proxectos.
- Fai *update* de ambos proxectos. Dará un aviso de que non pode mesturar e terás que decirlle cal é a modificación correcta

1.3.3 Tarefa 3. Instalación de TortoiseSVN nunha máquina virtual Windows 7 de 32 bits.

1.3.4 Tarefa 4. Utilización de Tortoise en local. Create repository here.

1.3.5 Tarefa 5. Utilización de Tortoise. Cliente e servidor na mesma máquina.