

Utilizando subversion como controle de versão

Heitor Rapcinski

Este documento descreverá a utilização de subversion como ferramenta de controle de versão mostrando as facilidades da sua utilização para administrar alterações em qualquer tipo de documento. Será mostrada a sua instalação em um servidor Windows e a utilização no desenvolvimento de aplicações Java com o Eclipse e com qualquer tipo de documentos utilizando TortoiseSVN.

Introdução

Subversion é um controle de versão free/open-source que permite armazenar documentos de qualquer natureza efetuando controle de acesso aos repositórios e mantendo as alterações através de um controle de revisões. Subversion pode ser usado para controlar alteração de diversos tipos de documentos, mas no momento estaremos voltados a demonstrar a sua utilização para melhorar o trabalho de desenvolvimento de software em equipes.

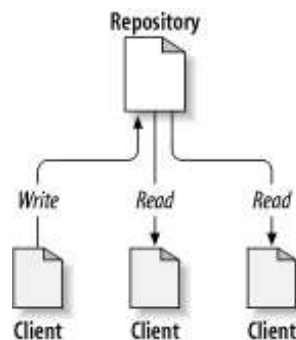
Utilizaremos nos exemplos o Eclipse com um plug-in chamado subclipse que pode ser encontrado através dos links apresentados abaixo.

Links:

Subversion <http://subversion.tigris.org/>
TortoiseSVN <http://tortoisesvn.tigris.org/>
Subclipse <http://subclipse.tigris.org/>
Eclipse <http://www.eclipse.org/>
SVNService <http://dark.clansoft.dk/~mbn/svnservice/>

Introdução

A arquitetura proposta pelo subversion é semelhante a um cliente-servidor, no servidor temos o repositório dos arquivos que são armazenados em um banco de dados e diversos clientes que são utilizados para acessar os documentos.



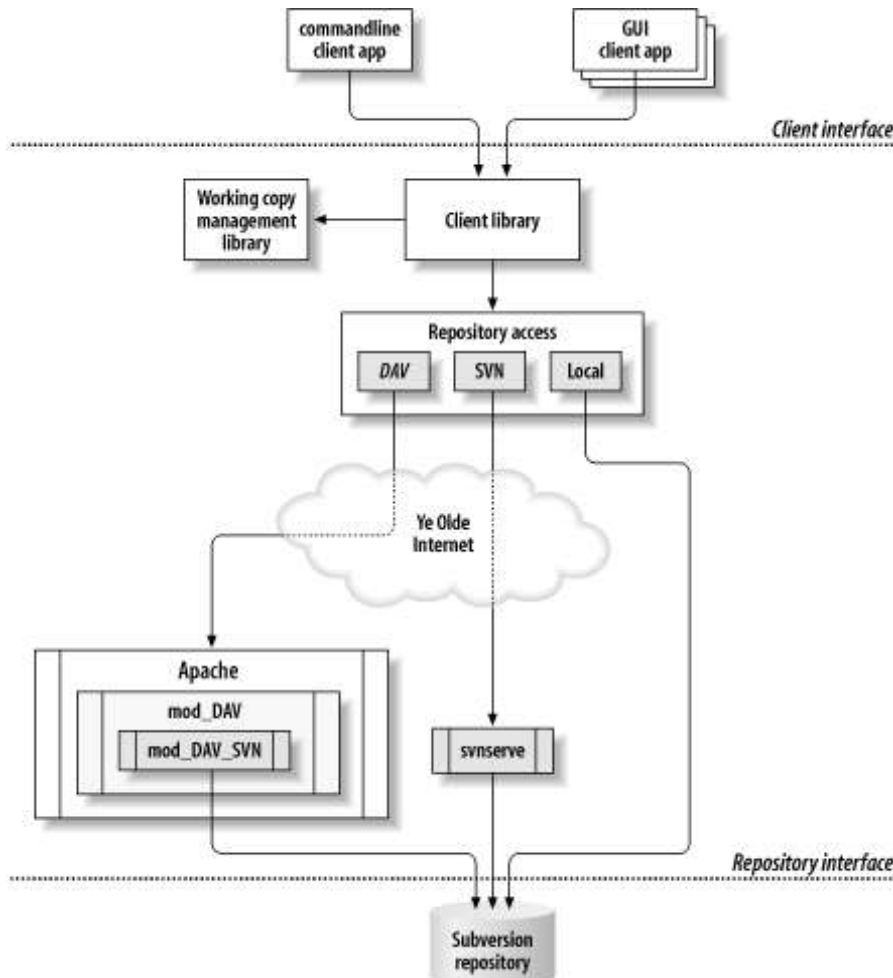
Os documentos são recuperados do repositório e armazenados localmente, após os documentos serem alterados pode-se submetê-los ao servidor, essa atividade altera o número da revisão do repositório. O número de revisão é sequencial iniciando em 1 e incrementada a cada alteração dos documentos. Cada documento tem armazenado no repositório sua última situação e a situação em cada uma das revisões, desta forma podemos reverter qualquer alteração, podemos comparar o documento com uma revisão específica.

Através do subversion podemos administrar melhor as alterações nos documentos com diversos recursos que essa ferramenta oferece. Todo o conteúdo do repositório fica registrado, assim podemos saber em que revisão o documento foi alterado, quem alterou. Fica armazenada no repositório a situação do documento em cada uma das revisões. Cada revisão possui a data, hora e usuário que a efetuou.

O acesso ao repositório pode ser anônimo ou através de um usuário autorizado, é definido para cada repositório um conjunto de usuários e senhas que tem acesso.

Podemos liberar impedir ou definir como somente de leitura tanto o acesso anônimo quanto o acesso através de usuário e senha.

O Subversion pode ser acoplado ao Apache e desta forma ficar disponível pela internet e até mesmo utilizar um protocolo seguro para o acesso aos documentos (https).



Instalação

Para instalar o servidor de subversion no Windows basta fazer o download do arquivo através do link e descompacta-lo em um diretório.

A partir do diretório <diretóriodeinstalacao>\bin podemos executar iniciar o servidor de subversion. Através da linha de comando a seguir estaremos iniciando o servidor e definindo o diretório dos nossos repositórios para a pasta "c:\repositoriosvn".

```
svnserve -d -r c:\repositoriosvn
```

Com o nosso servidor rodando devemos criar um repositório para armazenar os documentos, faremos isso através do comando `svnadmin`, criaremos então o repositório "aplicacao-java" em "c:\repositoriosvn\aplicacao-java".

```
svnadmin create c:\repositoriosvn\aplicacao-java
```

Devemos observar que os repositórios podem ser criados em qualquer pasta que esteja vazia, mas apenas poderão ser acessados os repositórios criados a partir do repositório principal definido através do comando `svnserve`. O comando `svnadmin` cria a estrutura básica do repositório que armazenará todos os dados.

Após a criação da estrutura inicial do repositório devemos editar o arquivo `svnserver.conf` que estará na pasta `\conf` do nosso repositório, seguindo nosso exemplo encontraremos esse arquivo na pasta "c:\repositoriosvn\aplicacao-java\conf". O conteúdo padrão do arquivo será:

```
### This file controls the configuration of the svnserve daemon, if you
### use it to allow access to this repository. (If you only allow
### access through http: and/or file: URLs, then this file is
### irrelevant.)

### Visit http://subversion.tigris.org/ for more information.

# [general]
### These options control access to the repository for unauthenticated
### and authenticated users. Valid values are "write", "read",
### and "none". The sample settings below are the defaults.
# anon-access = read
# auth-access = write
### The password-db option controls the location of the password
### database file. Unless you specify a path starting with a /,
### the file's location is relative to the conf directory.
### The format of the password database is similar to this file.
### It contains one section labelled [users]. The name and
### password for each user follow, one account per line. The
### format is
###   USERNAME = PASSWORD
### Please note that both the user name and password are case
### sensitive. There is no default for the password file.
# password-db = passwd
### This option specifies the authentication realm of the repository.
### If two repositories have the same authentication realm, they should
### have the same password database, and vice versa. The default realm
### is repository's uuid.
# realm = My First Repository
```

Para o nosso exemplo vamos modificar o conteúdo do arquivo para:

```
[general]
anon-access = read
auth-access = write
password-db = passwd
```

O termo **anon-access** define como será o acesso anônimo ao repositório, podemos definir como **none**, **read** ou **write** para impedir o acesso anônimo, permitir somente leitura ou permitir para leitura e gravação respectivamente. O termo **auth-access** define como poderá ser o acesso para usuários autorizados e pode ser definido com os mesmos valores **none**, **read** ou **write**. O termo **password-db** define o nome do arquivo contendo a lista de usuário e senha autorizados no repositório, no exemplo deveremos ter um arquivo chamado **passwd** a seguir um exemplo de um arquivo de senhas:

```
[users]
joao = 123
maria = abc
```

No nosso exemplo teremos um repositório que permite leitura dos documentos através de um acesso anônimo e leitura e gravação para os usuários autenticados. O acesso autorizado pode ser feito pelos usuários joão e maria através da senha 123 e abc respectivamente, é importante que tanto usuário como senha são case sensitive. Vários outros parâmetros podem ser adotados para melhorar a segurança do repositório, mas não trataremos disso agora.

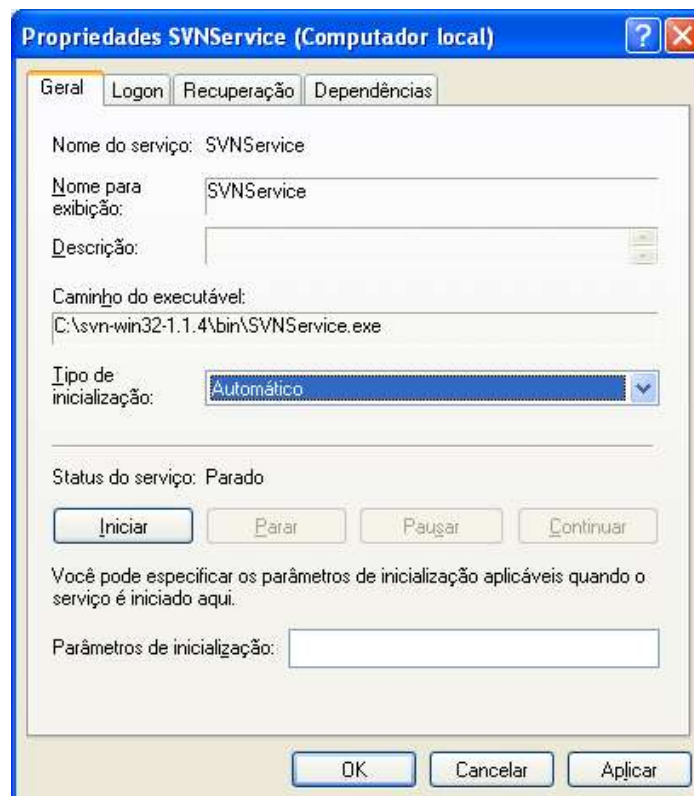
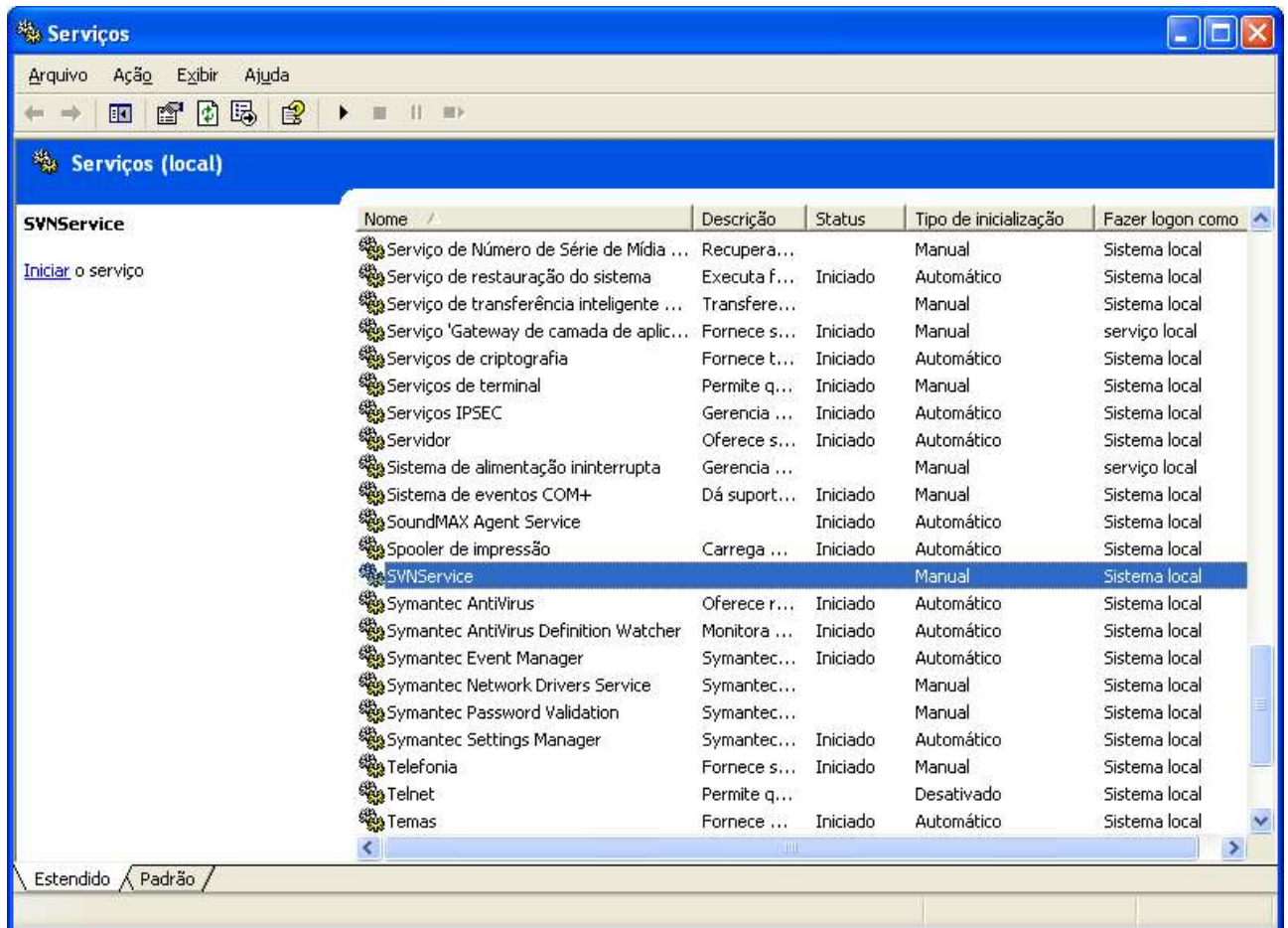
Configurando o svnserve como serviço do windows

O servidor de subversion já está funcionando, enquanto o comando svnserve estiver em execução o servidor estará no ar, mas em se tratando de um servidor Windows qualquer ação que feche o prompt de comando fará com que o serviço seja interrompido.

Para resolver esse incomodo podemos utilizar o SVNService, que pode ser conseguido através do link relacionado no inicio deste artigo. O SVNService registra o svnserve na lista de serviços do Windows. Basta descompactar o conteúdo do arquivo e copiar os arquivos para o diretório \bin do subversion. Para instalar o subversion como serviço:

```
svnservice -install -d -r c:\repositoriosvn
```

Através do comando anterior colocaremos o svnserve rodando como um serviço do windows, podemos então controla-lo através do gerenciador de serviços do windows.



Podemos editar o tipo de inicialização para automático e iniciar o serviço assim mesmo que o servidor seja reiniciado o serviço de subversion será reativado.

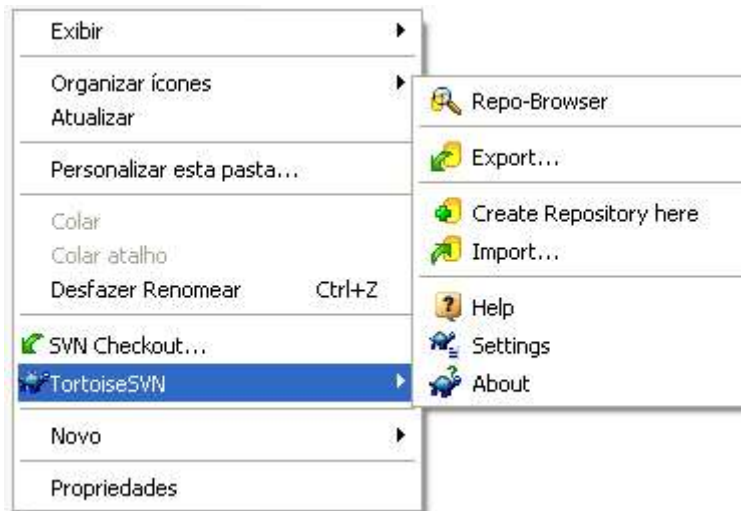
Para remover o serviço basta executar o seguinte comando:

```
svnservice -remove
```

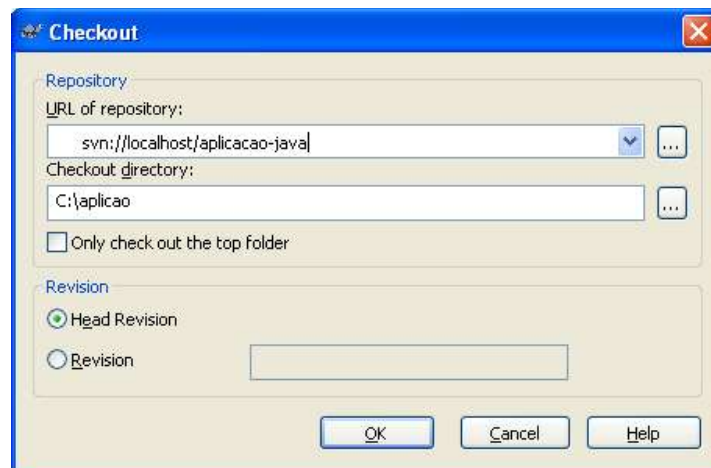
Usando o TortoiseSVN

O TortoiseSVN é um cliente para subversion que permite acessar um repositório integrando os documentos diretamente no Windows Explorer.

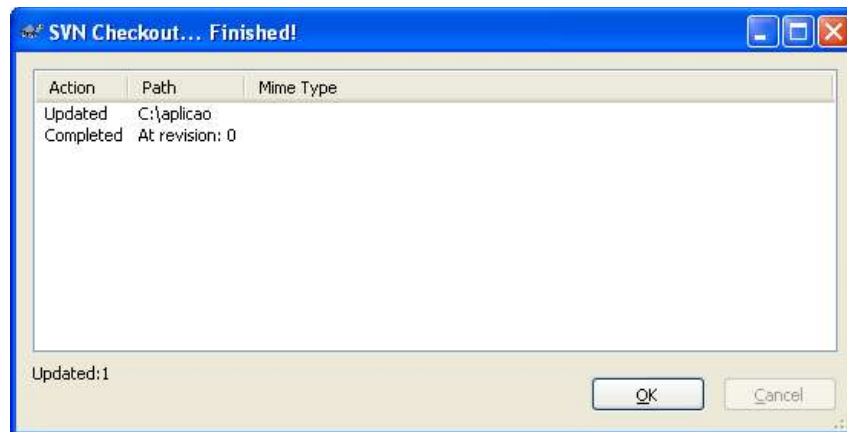
Faça download a partir do link e faça a instalação, após a instalação aparecerá no menu de contexto de qualquer pasta as opções do TortoiseSVN.



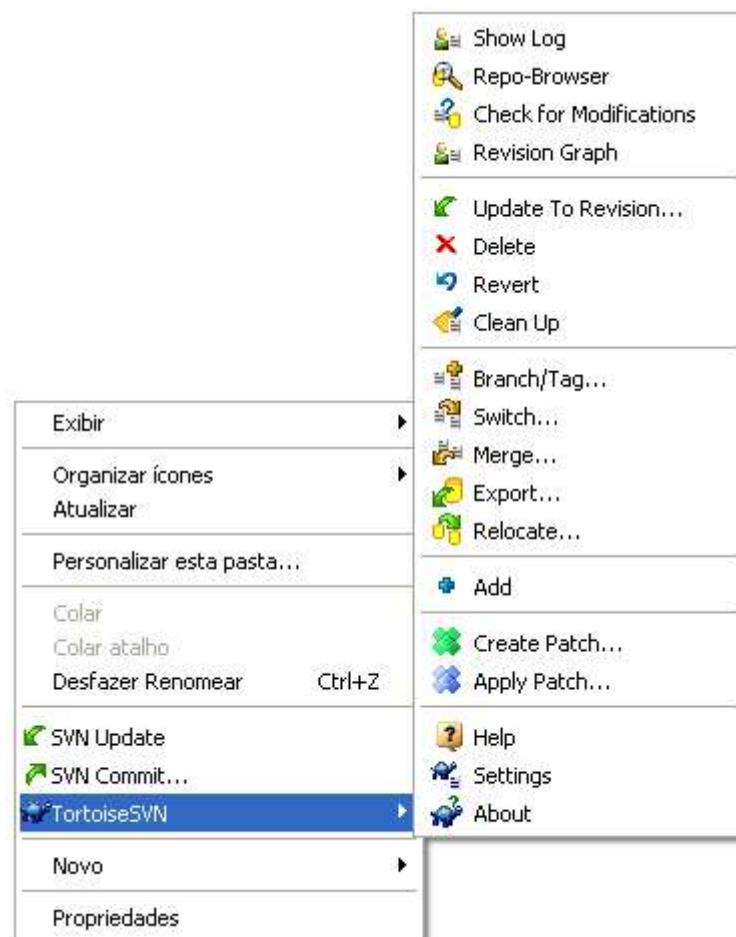
Então basta criar uma pasta e executar um SVN Checkout que vai fazer com que os documentos do repositório seja trazidos para a sua pasta local. Utilizaremos o repositório "aplicação-java" criado nos exemplos anteriores. Veja que utilizaremos o `svn://localhost/aplicacao-java` como URL, isso vai variar de acordo com sua configuração de servidor e arquitetura de instalação do subversion. No checkout podemos escolher a revisão ou utilizar HEAD para que seja trazida a revisão mais atual do repositório.



De acordo com a configuração do repositório será solicitado um usuário e senha mesmo para a leitura dos documentos, isso dependerá com a configuração do arquivo `svnserve.conf`. Após o checkout será exibida a tela com o resumos dos documentos trazidos no checkout.

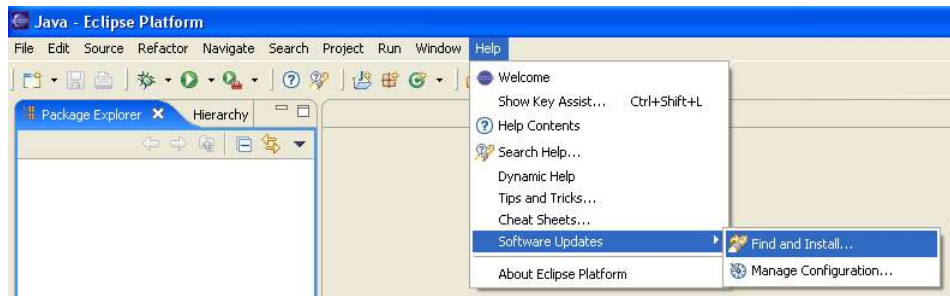


Após ter feito checkout de um repositório para uma pasta, sempre que for acessado o menu de contexto dentro da pasta aparecerão todas as operações que poderemos realizar no repositório. As duas principais funções são: SVN Commit que grava todos os dados da pasta local para o repositório e SVN Update que traz os documentos do repositório para a pasta local.

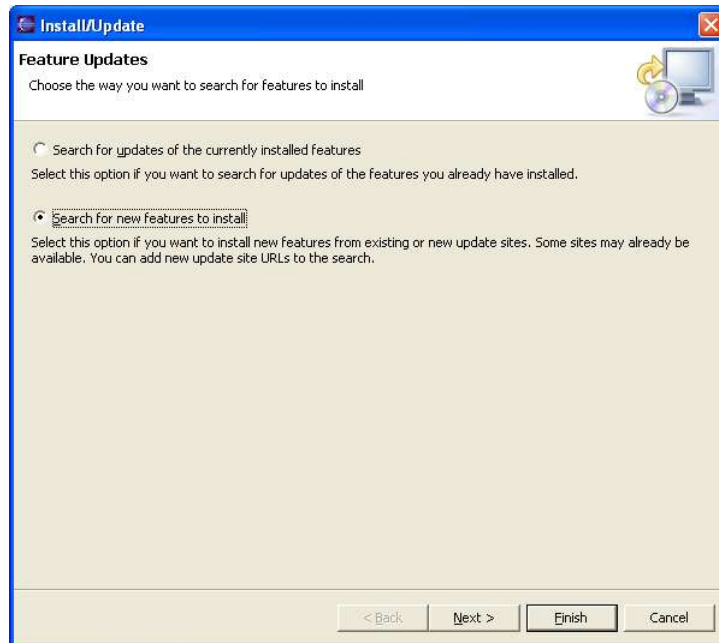


Instalando o Subclipse no Eclipse

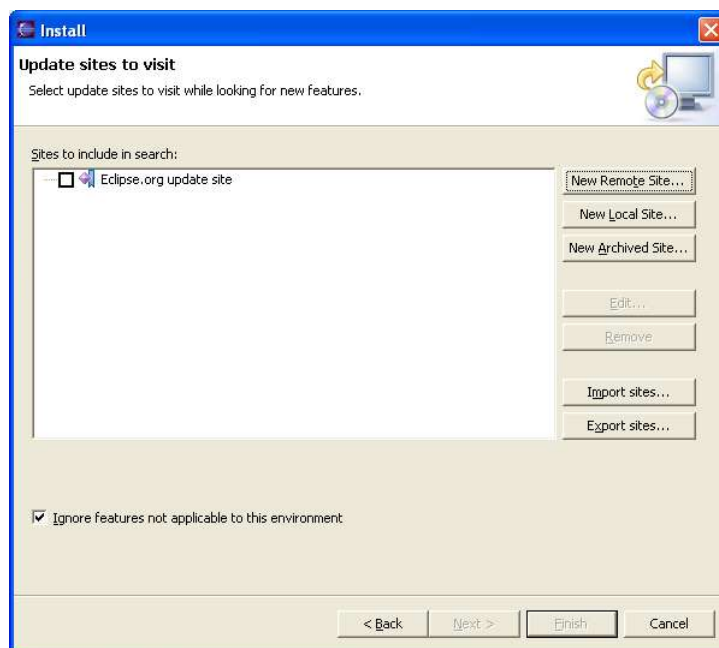
O Subclipse é um plug-in que é instalado no Eclipse que permite realizar todas as funcionalidades do subversion através do Eclipse. Para instalar o plug-in no Eclipse faça o seguinte procedimento:



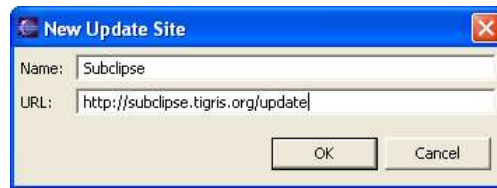
Entre na Opção Help->Software Updates->Find and Install



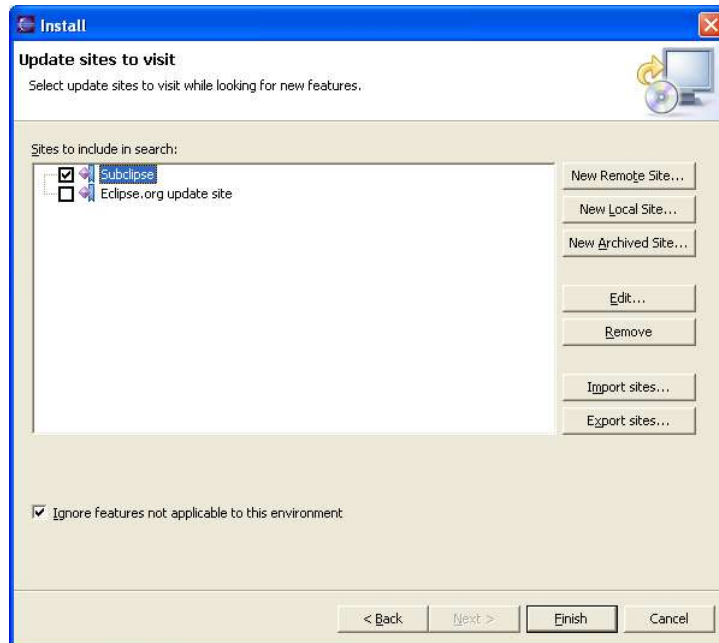
Selecione Search for new features to install



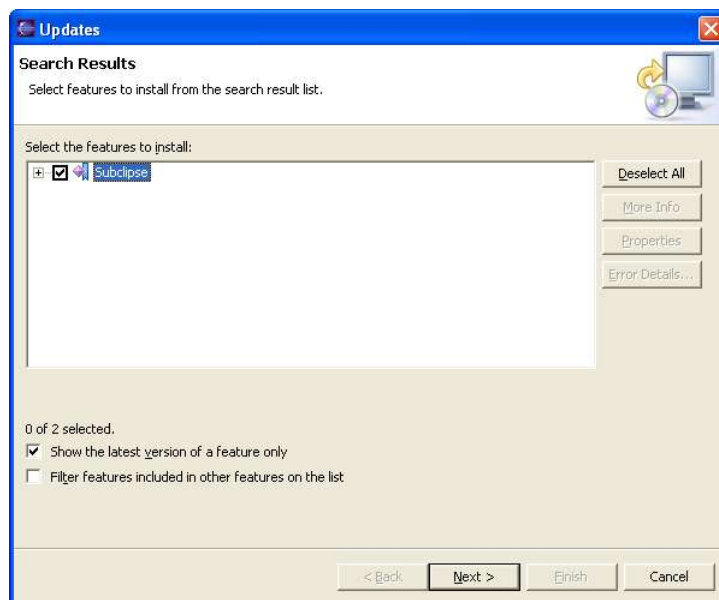
Clique em New Remote Site



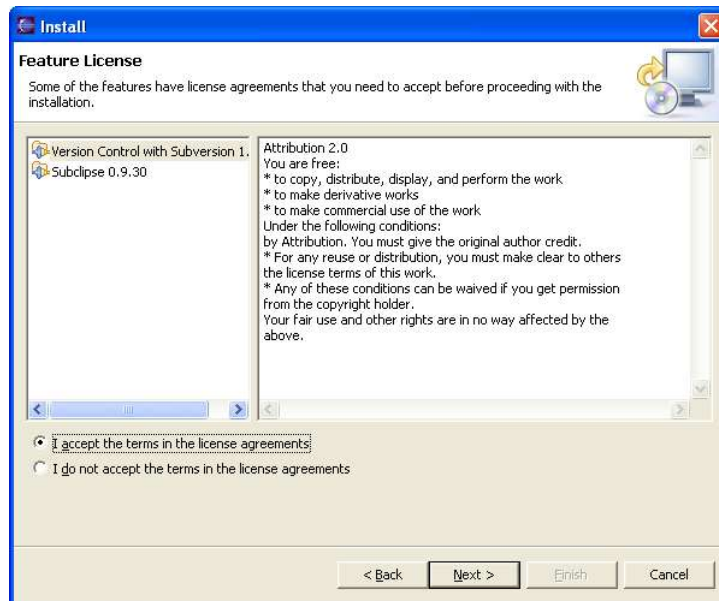
Preencha com os dados do site



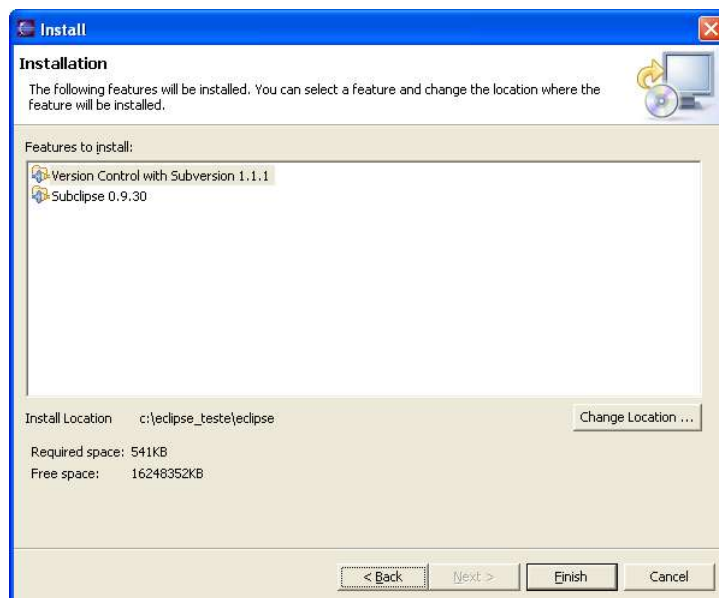
Selecione Subclipse e clique em Finish



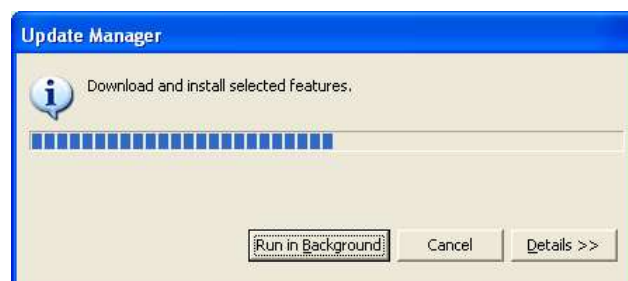
Selecione Subclipse e clique em Next



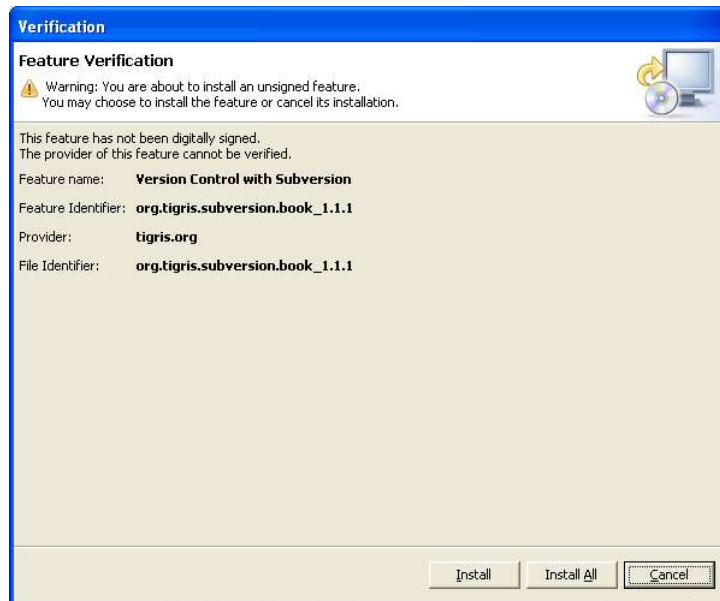
Coloque em I accept... e clique em Next



clique em Finish



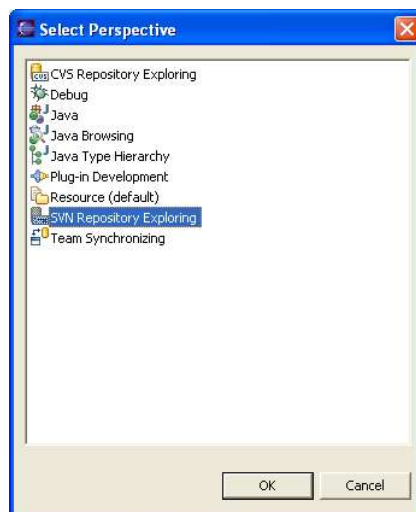
Aguarde o download



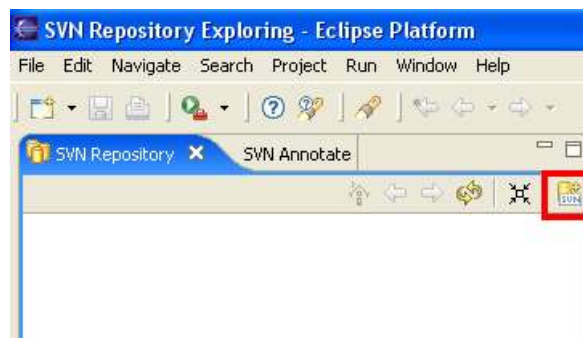
Clique em Install All

Usando o Subclipse

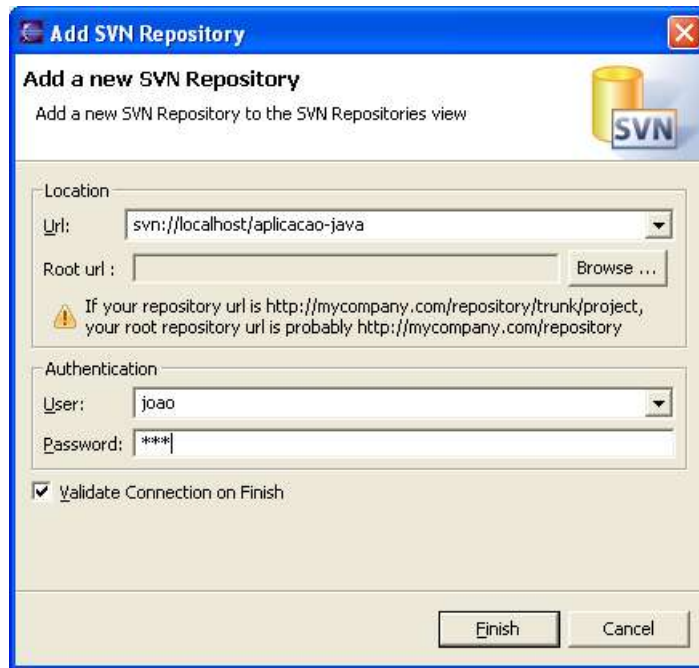
Com o subclipse instalado no Eclipse podemos definir a conexão com o nosso servidor, vamos utilizar o mesmo exemplo citado anteriormente. No Eclipse seleciona a perspectiva SVN Repository Explorer.



Selecione a Perspectiva SVN Repository Explorer

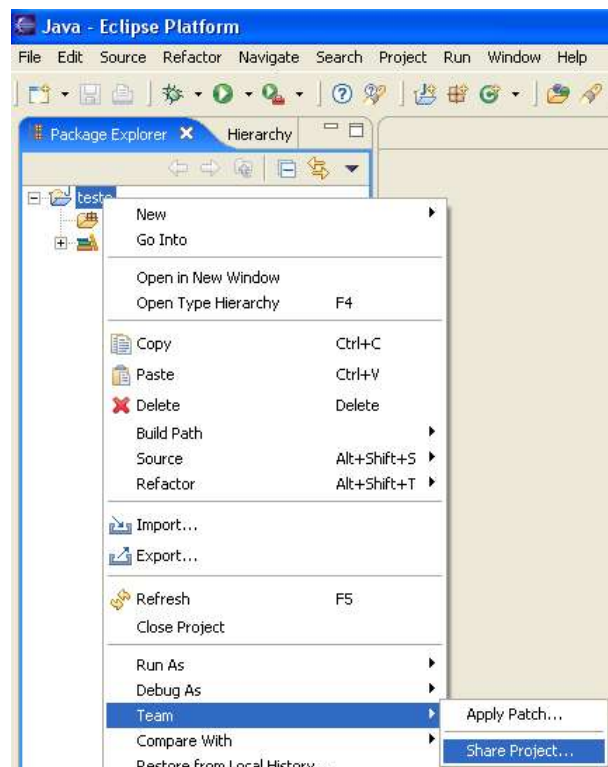


Clique em Add SVN Repository

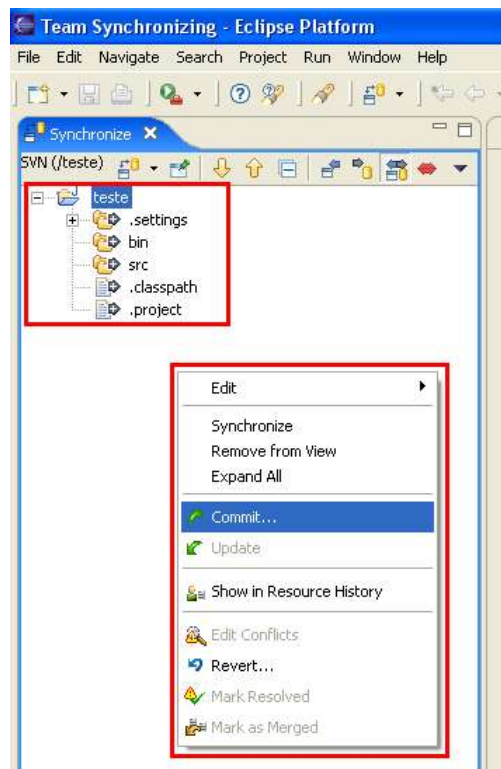


Preencha com os dados do Repositório

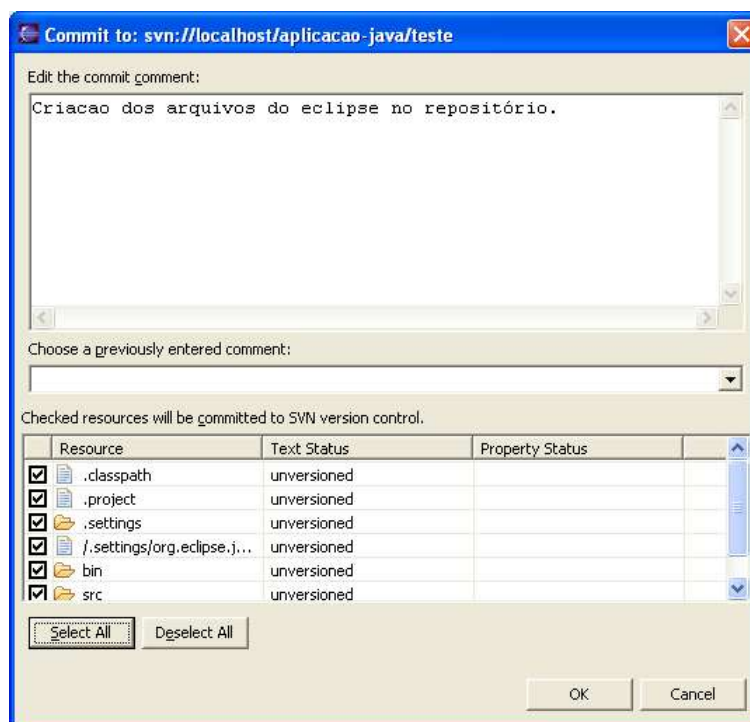
Agora criamos no Eclipse uma conexão para o nosso repositório, agora basta criar um projeto no Eclipse e compartilhá-lo através do subclipse. Faça a criação de um projeto qualquer, clique com o botão direito nele e execute Team->Share Project... siga os passos selecionando o repositório.



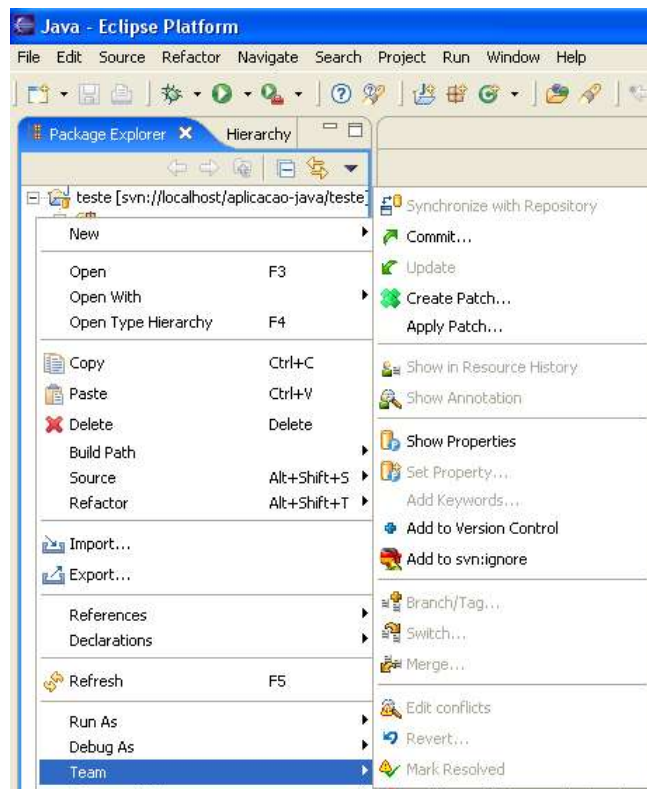
Após terminar o processo de compartilhamento do seu projeto o Eclipse indicará o processo de sincronização entre o seu projeto local e o repositório. A perspectiva Synchronize facilita o processo, são mostrados todos os documentos que serão adicionado no repositório e através do menu de contexto é possível confirmar o envio dos documentos para o repositório.



Quando é executado o comando commit o usuário pode definir quais arquivos serão enviados e também preencher um campo com os comentários sobre a atualização.



Depois que for realizada a sincronização com o repositório o Eclipse estará fazendo sempre um controle das alterações do projeto e através da opção Team é possível realizar todas as operações de subversion entre o Eclipse e o repositório. O Eclipse fará com que os ícones de cada arquivo do seu projeto fique decorado de tal forma que seja possível identificar a data e hora, usuário e revisão do arquivo no repositório.



Conclusão

Subversion é uma ótima ferramenta para administração de projetos com uma equipe de desenvolvedores, demonstramos nesse artigo um passo inicial para instalar e trabalhar no Eclipse com Java ou com qualquer outro tipo de documento.

Heitor Rapcinski (heitorr@yahoo.com, heitor_10@hotmail.com).