

CVS – Controle de versão

Conheça uma ferramenta free e integrada ao Windows Explorer



CRISTIANO CAETANO

(c_caetano@hotmail.com)

autor do livro CVS: Controle de Versões e Desenvolvimento Colaborativo de Software. Com 10 anos de experiência, já trabalhou na área de qualidade e teste de software para grandes empresas como Zero G e DELL. Atualmente está exercendo a função de Software Test Engineer na HP Invent. O autor também pode ser contactado no seguinte endereço: spaces.msn.com/softwarequality.

O CVS é uma ferramenta *open source* que implementa as principais funções pertinentes ao processo de controle de versões. Basicamente, o CVS armazena em seu repositório as modificações realizadas num arquivo ao longo do tempo; cada modificação é identificada por um número chamado *Revisão*.

Toda *Revisão* armazena as modificações realizadas, quem realizou as modificações, quando foram realizadas, entre outras informações. Além disso, o CVS conta com um mecanismo capaz de controlar os acessos simultâneos e as modificações paralelas, garantindo a integridade das modificações e a atomicidade das operações.

De maneira geral, o fluxo de trabalho básico do CVS consiste em retirar os arquivos de um projeto para a área de trabalho, realizar as modificações e, por último, transferir as modificações ao repositório, como pode ser visto na **Figura 1**.

No entanto, durante o curso das modificações, você poderá transferir as suas modificações ao repositório ou obter as modificações realizadas por outras pessoas. O completo entendimento dos termos apresentados na **Tabela 1** é fundamental para auxiliar a leitura dos próximos tópicos.

TortoiseCVS

O TortoiseCVS é um *front end* gráfico que torna o uso do CVS mais fácil e intuitivo. Basicamente, o TortoiseCVS realiza as principais operações do CVS por meio de menus de contexto integrados ao Windows Explorer, como pode ser visto na **Figura 2**.

O TortoiseCVS é, em virtude da sua proposta, uma ferramenta extremamente eficiente cuja principal característica é a sua independência da linguagem de programação. Em razão disso, você poderá utilizar o TortoiseCVS sem precisar inserir nenhum *plugin* no IDE do Delphi. Dessa

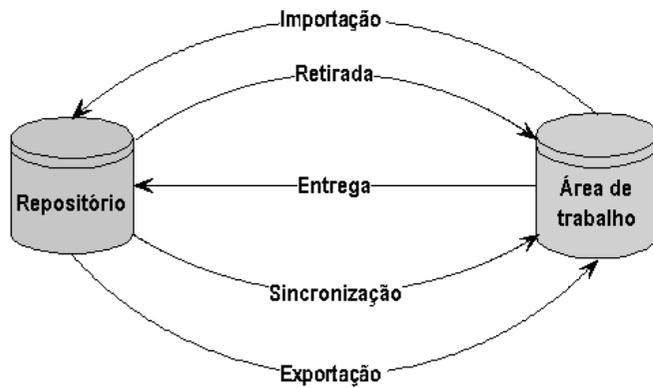


Figura 1. Operações realizadas pelo CVS

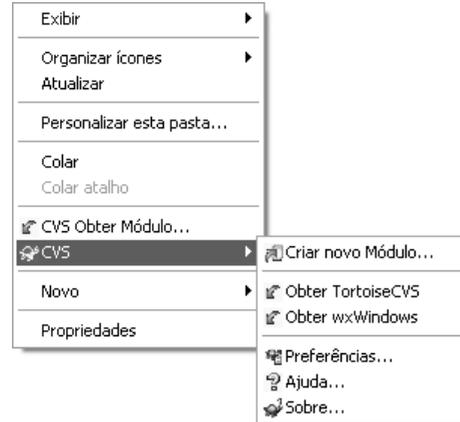


Figura 2. Menus de contexto integrados ao Windows Explorer

Termo	Descrição
Workspace, SandBox (Área de trabalho)	Termo usado para representar um diretório no computador cliente onde os arquivos de um projeto serão transferidos durante uma retirada.
Repository (Repositório)	Local onde os arquivos submetidos ao controle de versões são armazenados.
Revision (Revisão)	Termo usado para descrever a numeração atribuída pelo CVS a cada modificação de um arquivo.
Release, Version (Liberação)	Termo usado para descrever a denominação atribuída a um conjunto de arquivos para identificar determinado ponto no tempo, sobretudo, quando se quer identificar um conjunto de novas características ou correções de um software.
Commit (Entregar, Submeter)	Termo usado para descrever a transferência de um ou mais arquivos modificados da Área de Trabalho para um Repositório.
Checkout (Retirar, Recuperar)	Termo usado para descrever a retirada de um ou mais arquivos de um repositório para a área de trabalho.
Diff (Comparação das diferenças)	Termo usado para descrever o processo de comparação das modificações entre revisões diferentes de um arquivo.
Import (Importação)	Termo usado para descrever o processo de transferência dos arquivos de um projeto a um repositório, dando início ao controle de versões.
Module (Projeto, Módulo)	Termo usado para descrever um nome simbólico associado a um conjunto de diretórios e arquivos que compõem um projeto, a fim de facilitar o acesso e a manipulação.
Branch (Ramo)	Termo usado para descrever o processo de divisão dos arquivos de um projeto em linhas de desenvolvimento independentes.

Tabela 1. Termos utilizados no controle de versões

forma, como descrito anteriormente, todas as operações serão realizadas por meio de menus de contexto do Windows Explorer.

O TortoiseCVS é uma ferramenta gratuita distribuída sob a licença GPL. Segundo o site do desenvolvedor, ele roda normalmente nas seguintes versões do Windows: 95, 98, ME, NT, 2000 e XP.

Dica: Na edição 73 também apresentamos uma outra alternativa gratuita para controle de versão, confira.

Repositório

Basicamente, um repositório é um diretório que tem por objetivo abrigar todos os arquivos de um projeto sob o controle de versões. O repositório poderá ser criado num diretório compartilhado no seu computador ou num servidor (Windows ou Linux). Naturalmente, quando o repositório é criado num servidor, você deverá determinar um método de acesso ao repositório para garantir a segurança dos arquivos sob o controle de versões.

Sendo assim, para fins didáticos, os exemplos apresentados nas seções seguintes serão realizados num repositório local. Para tal tarefa, você deverá criar e compartilhar um diretório chamado "CVSROOT" na raiz do seu computador (C:\CVSROOT).

Importação

Uma vez que o repositório tenha sido criado, o primeiro passo para iniciar o controle de versões é chamado de importação. Nesse passo, os arquivos da sua aplicação Delphi são transferidos para o repositório e submetidos ao controle de versões do CVS. Para tal tarefa, devemos copiar os arquivos do projeto numa área temporária a fim de realizarmos a importação dos arquivos. Para fins de análise e entendimento, vamos transferir os arquivos de um suposto projeto de um software chamado *Calculadora* para o diretório "C:\Temp".

A importação do projeto deverá ser realizada por meio do menu de contexto *Criar Novo Módulo*. Na caixa de diálogo, o TortoiseCVS fornece opções para acessar repositórios remotos por meio de qualquer método de autenticação suportado pelo CVS. Mas, no entanto, como havíamos definido anteriormente, todos os exemplos serão realizados utilizando um repositório local. Em razão disso, você deverá selecionar a opção *Diretório Local (:local:)* do item *Protocolo*.

O campo *Pasta do Repositório* deverá ser preenchido com "C:\CVSROOT" que é a localização do repositório criado anteriormente na raiz do seu computador. O campo *Módulo* indica um apelido que será utilizado pelo TortoiseCVS todas as vezes que ele se referir ao nosso projeto e deverá ser preenchido com o texto "CALC", como pode ser visto na **Figura 3**.

Após confirmar os dados, o TortoiseCVS exibirá um diálogo informando que não existe nenhum repositório criado no diretório informado. Você deverá selecionar a opção para inicializar um novo repositório, como pode ser observado na **Figura 4**.

Esse diálogo será exibido somente na primeira vez que você importar o projeto em função de que o repositório não tinha sido inicializado ainda. Posteriormente, assim que a criação do novo módulo e a inicialização do repositório forem completadas, você deverá adicionar os arquivos a esse módulo recém criado. Para tal tarefa, você deverá acessar o menu de contexto *CVS Adicionar Conteúdo*.

O TortoiseCVS exibirá um diálogo solicitando a confirmação dos arquivos que serão adicionados. Adicionalmente, você poderá informar o formato do arquivo, o tipo de final de linha do arquivo, entre outras opções, como pode ser visto na **Figura 5**.

Por fim, devemos realizar a operação de entrega, também chamada de *Commit* para transferir os arquivos do projeto definitivamente para o repositório do CVS. Essa operação deverá ser realizada através do menu de contexto *CVS Gravar*

O TortoiseCVS exibirá um diálogo solicitando a confirmação dos arquivos que serão transferidos para o repositório. Neste diálogo, você também deverá escrever um breve comentário descrevendo as modificações realizadas. Além disso, antes de confirmar a operação você poderá modificar o formato do arquivo, comparar as suas modificações com outras versões armazenadas no repositório, entre outras opções, como pode ser observado na **Figura 6**.

Retirada dos arquivos do repositório

Assim que os arquivos de um projeto forem importados com sucesso, podemos finalmente iniciar as operações de controle de versões. Para tal tarefa, devemos retirar os arquivos armazenados no repositório e transferi-los para a área de trabalho, onde poderemos realizar as modificações nos arquivos do projeto CALC.

A área de trabalho é um diretório qualquer no disco rígido. Nesse caso, vamos assumir que a nossa área de traba-



Figura 3. Caixa de diálogo para criar um novo módulo

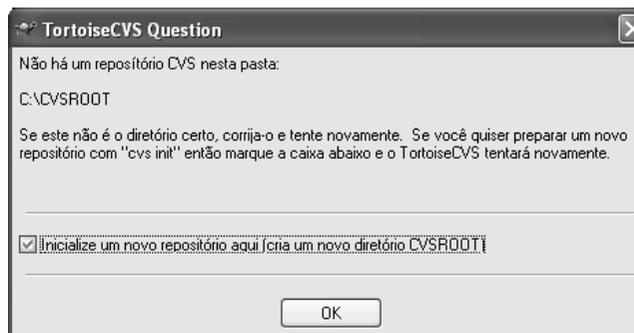


Figura 4. Inicialização do repositório

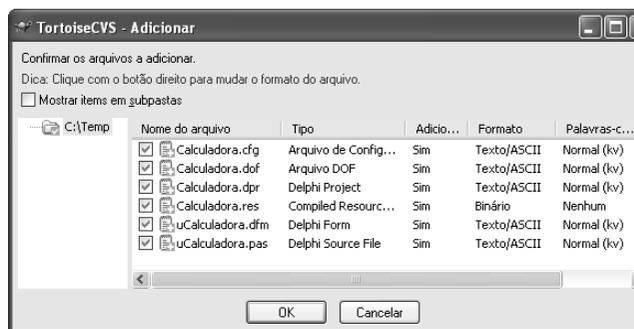


Figura 5. Adicionando arquivos ao módulo

ho será o diretório "C:\Projetos". Na prática, para realizar essa tarefa, você deverá clicar sobre o diretório (*Projetos*) e acessar o menu de contexto *CVS Obter Módulo*. Como resultado, o TortoiseCVS exibirá o diálogo para realizar a retirada dos arquivos, também conhecido como *Checkout*.

Por meio desse diálogo, você poderá configurar onde está localizado o repositório de arquivos, qual projeto você deseja retirar, entre outras opções, como pode ser visto na **Figura 7**.

Cabe lembrar que a não ser que você faça alguma configuração adicional, o TortoiseCVS armazena uma lista dos projetos/repositórios acessados anteriormente, facilitando assim posteriores retiradas. Por fim, assim que a operação de retirada for concluída, você poderá começar a trabalhar com os arquivos do projeto CALC que foram transferidos para o diretório *Projetos*.

Modificando os arquivos

Assim que o projeto for transferido para a área de trabalho, poderemos iniciar a edição dos arquivos. Perceba que após a importação e durante a edição dos arquivos, o TortoiseCVS modifica os ícones dos arquivos a fim de dar um significado visual sobre o seu status em relação ao repositório, como pode ser visto na **Figura 8**.

A rigor, a edição dos arquivos do seu projeto não mudará em nada. Utilize o Delphi da mesma maneira que você está acostumado. No entanto, para que a adição e remoção de arquivos sejam entendidas pelo CVS, você deverá selecionar o arquivo desejado e acessar os menus de contexto *CVS Adicionar* ou *CVS Remover* como mostrado na **Figura 9**.

Por fim, devemos lembrar de realizar a operação de entrega (*Commit*) ao final do nosso trabalho para garantir que todas as modificações, adições e remoções sejam definitivamente refletidas no repositório do CVS.

Configurações Adicionais

O TortoiseCVS em virtude da sua arquitetura, ou melhor, da arquitetura do CVS, adiciona automaticamente um *linefeed* (#10) antes de cada caractere *carriage return* (#13) dentro dos arquivos armazenados em seu repositório. Na verdade, ele só realiza esse tipo de operação em arquivos definidos com o formato texto puro (*plain text*).

O problema é que ele identifica por default os arquivos DFM como se fossem arquivos texto puro, corrompendo os mesmos durante o armazenamento desses arquivos no repositório CVS. Para sanar esse problema, você deverá selecionar individualmente cada arquivo DFM e mudar o campo *Formato* de *Text/ASCII* para *Binary* quando realizar a operação de entrega (*Commit*) do arquivo.

Como esse procedimento é trabalhoso e sujeito a erros, você poderá configurar o TortoiseCVS de modo que ele possa automaticamente identificar todos os arquivos DFM como se fossem arquivos binários. Em resumo, existe um arquivo de configuração utilizado pelo TortoiseCVS onde poderão ser definidos os formatos dos arquivos conforme a sua extensão ou por uma expressão regular.

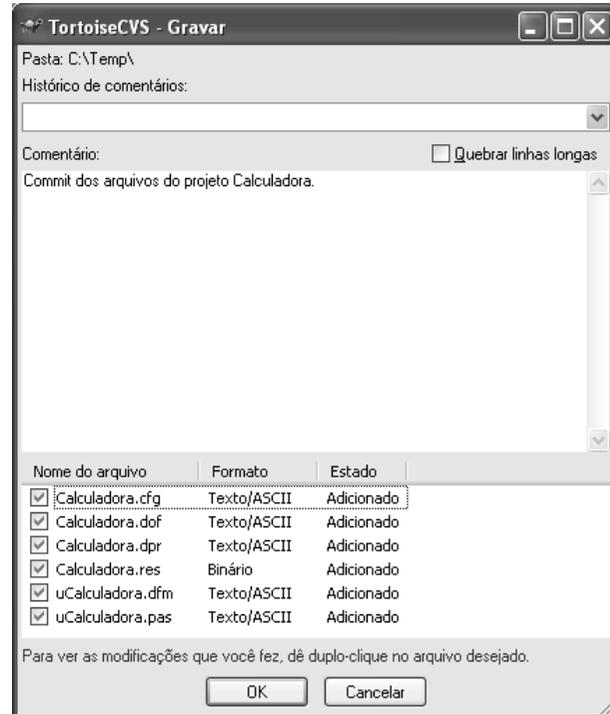


Figura 6. Transferência definitiva dos arquivos ao repositório CVS

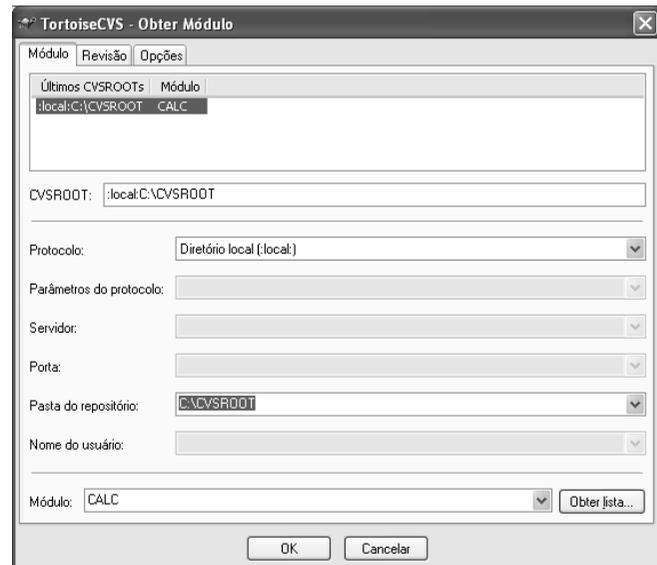


Figura 7. Retirada dos arquivos do repositório CS

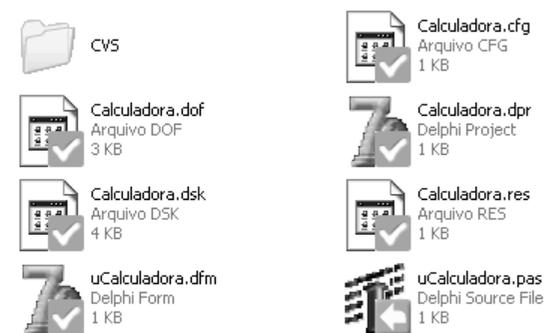


Figura 8. Ícones modificados indicando o status atual dos arquivos

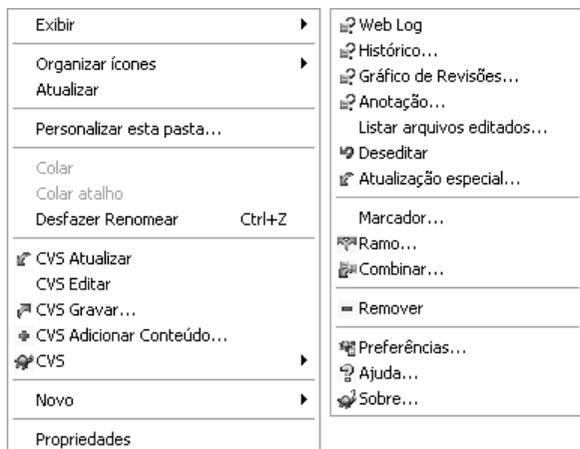


Figura 9. Adicionando e removendo arquivos

Para realizar tal tarefa, você deverá editar o arquivo *TortoiseCVS.FileTypes* (utilizando o Bloco de Notas) que está localizado na raiz da instalação do TortoiseCVS (normalmente é o diretório C:\Arquivos de programas\TortoiseCVS) e inserir a expressão “*.dfm binary” no final do arquivo, como pode ser visto na **Figura 10**.

Normalmente arquivos executáveis e arquivos DCU não deverão ser submetidos ao controle de versões. Via de regra, arquivos binários resultantes da compilação, normalmente não deverão ser submetidos ao controle de versões.

Em razão disso, antes de realizar a importação ou adição de arquivos, você deverá configurar o TortoiseCVS para ignorar os arquivos que não deverão ser submetidos ao controle de versões. Para tal tarefa, você deverá acessar a guia *Arquivos Ignorados* do diálogo *Preferências* e adicionar os arquivos que deverão ser ignorados, como pode ser visto no exemplo descrito na **Figura 11**.

Conclusão

Neste artigo foram apresentadas alguns dos recursos oferecidos pelo TortoiseCVS. Com base nessas informações você já estará apto para iniciar o controle de versões dos seus projetos, aumentando assim a confiabilidade e a rastreabilidade do seu trabalho. Se você desejar conhecer os demais recursos oferecidos pelo CVS, vale a pena dar uma lida no livro *CVS: Controle de Versões e Desenvolvimento Colaborativo de Software* (veja a seção *Links*). ■



Figura 10. Definindo automaticamente o formato dos arquivos DFM

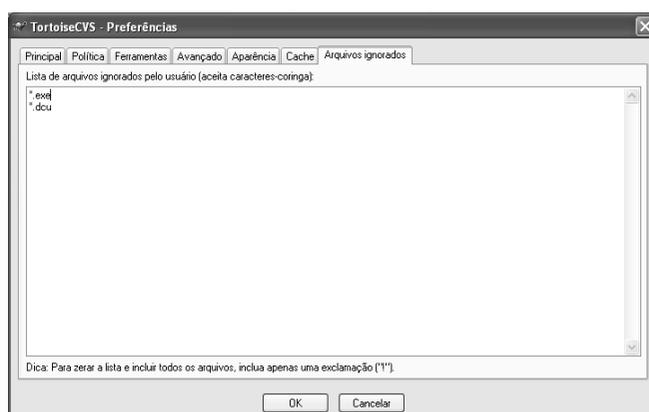


Figura 11. Definindo os arquivos que deverão ser ignorados

Links

TortoiseCVS homepage

www.tortoisecvs.org

Resolving DFM corruption issues with CVS

bdn.borland.com/article/0,1410,26068,00.html

CVS: Controle de Versões e Desenvolvimento Colaborativo de Software

www.novateceditora.com.br/livros/cvs

Informações sobre o CVS

ximbiot.com/cvs/wiki/index.php?title=Main_Page

Blog do autor contendo informações adicionais sobre CVS e qualidade de software

spaces.msn.com/softwarequality