Utilización de WinCVS

Un taller, para comenzar

Jaime Irving Dávila Latorre

Juan Pablo Quiroga González

Utilización de WinCVS: Un taller, para comenzar
por Jaime Irving Dávila Latorre y Juan Pablo Quiroga González

Historial de revisiones
Revisión 1.02001-04-03Revised by: jid
Creación de documento inicial usando texinfo

Historial de revisiones
Revisión 1.12001-05-11Revised by: jid
Formato Docbook. Sugerencias de Vladimir Tamara y Carlos Julio Ruiz.

Historial de revisiones
Revisión 1.22001-08-06Revised by: jpq
Formato Docbook. Adición de configuración del servidor cuando se utiliza xinetd. Algunos comentarios cuando s

Historial de revisiones
Revisión 2.02002-01-24Revised by: jid
Énfasis en el uso de ssh, definición de depósitos de lectura usando cvspserver.

Tabla de contenidos

1. Introducción 4

2. Configuración del servidor Linux..........6

2.1. Creación del depósito central..........6

2.2. Creación del archivo de usuarios 7

2.3. Información para administradores 8

2.3.1. Si dispone de inetd[4] (Redhat 6.2 y anterior) 9

2.3.2. Si dispone de xinetd[5] (Redhat 7.0 y posterior)..........9

2.3.3. Información para la distribución Debian 10

3. Configuración del cliente en Windows..........12

3.1. Definición del depósito central 12

3.2. Ingreso al sistema..........13

3.3. Creación de un espacio de trabajo 14

3.4. Extracción de un módulo 16

3.4.1. Selección del módulo..........16

3.4.2. Edición de los archivos del módulo. 17

3.5. Actualización del depósito central. 19

3.5.1. Actualizacion en ausencia de conflictos de versiones..........19

3.5.2. Actualización en caso de conflictos de versiones 20

3.6. Uso de los *tag's*..........22

3.6.1. Creacion de los *tag's*..........22

3.6.2. Observación de los *tag's* disponibles..........23

3.6.3. Extracción de los archivos asociados con un *tag* 24

3.7. Salida de la sesión..........25

Índice..........26

Capítulo 1. Introducción

cvs (*concurrent version system*) es una herramienta que permite manejar de forma concurrente las versiones con un conjunto de archivos y directorios. Dichos archivos pueden corresponder a documentos (como es el caso de este documento) o archivos fuente de una determinada aplicación o programa. Este tipo de aplicación corre en un servidor (en este caso un servidor Linux) y permite que muchos clientes se conecten a ella, obteniendo copias de los archivos sobre los cuales se está trabajando. Quizás la forma más fácil de visualizar a cvs es como un servidor ftp (*file transfer protocol*) que además permite el manejo de versiones.

Hay multitud de programas usados por los clientes para conectarse con el depósito central cvs, como es el caso de la línea de comandos de cvs (probablemente si usted está corriendo Linux en su cliente esta es una de las más conocidas) y las interfaces gráficas. Con el objetivo de desarrollar clientes gráficos para una multitud de plataformas existe el proyecto guicvs (guicvs.org), que ha generado hasta el momento interfaces gráficas para Windows (wincvs, la aplicación que describiremos en este taller), Macintosh (maccvs) y Linux (gcvs). Este taller se enfoca en presentar un ejemplo de la funcionalidad de un servidor cvs en Linux, y de su cliente en Windows™, wincvs, (aunque por ejemplo, la interfaz gcvs es muy similar) en forma de taller.

Por ende, nuestros objetivos en este taller son:

- Crear un depósito central cvs para la administración de configuraciones.
- Aprender el uso del *checkin*(registro), *checkout*(extracción), en un depósito cvs.
- Manejar el uso de *tags*(marquillas).

Sin embargo no describimos en detalle toda las capacidades de cvs y wincvs, para ello remitimos a la documentación en línea de (wincvs (http://www.wincvs.org)). Sin embargo, siéntase libre de contactar a Jaime Irving (mailto: jadavila@uniandes.edu.co) o a Juan Pablo (mailto: jquiroga@uniandes.edu.co) en caso de imprecisiones en alguna de las secciones.

Esta versión (wincvs.sgml) de este documento fue realizado usando la herramienta

DocBook (http://www.docbook.org), en contraste de la primera versión (wincvs.texinfo) que fue realizada usando texinfo (http://www.texinfo.org). Queremos agradecer las valiosas sugerencias de Vladimir Támara (mailto: vtamara@tamarapatino.com) y Carlos Julio Ruiz (mailto: carlosjulioruiz@hotmail.com) en especial en Sección 2.3>.

Capítulo 2. Configuración del servidor Linux.

La configuración del servidor de cvs, depende del método que se usará después para conectarse con éste. En esta documentación básicamente estudiaremos dos formas de comunicarse con el servidor, ssh y un protocolo propio de cvs llamado pserver. El primero es el método preferido para depósitos de escritura y lectura, por cuanto los *passwords* viajan de forma segura por la red y además por cuánto require menos labores de administración, sin embargo tiene la desventaja de que se necesita tener un cliente de ssh en el cliente (en nuestro caso usaremos el ssh que viene con cygwin). El segundo es un método apropiado para depósitos de lectura solamente, su uso en depósitos de lectura/escritura puede ser un riesgo de seguridad, por cuanto las claves del usuario viajan por la red sin la debida encriptación y además la seguridad del sistema depende de la correcta administración del directorio `CVSROOT`. A continuación describimos el conjunto de pasos que son necesarios para configurar un servidor cvs, con sistema operacional Linux.

2.1. Creación del depósito central

- Inicie una sesión en el servidor donde desea ubicar el depósito central de su proyecto [1]
- Defina la variable CVSROOT en tal máquina, dicha variable, contendrá el nombre del directorio donde usted tendra su depósito, en nuestro ejemplo usaremos *`~/deposito`*. Para ello es necesaria la siguientes línea:

  ```
  [is25201@naoma is25201]$ exportCVSROOT=~/deposito
  ```

- Inicialice cvs, de la siguiente forma:

  ```
  [is25201@naoma is25201]$cvsinit
  ```

Después de este paso se creará un directorio de nombre *`deposito`*, en su directorio. Si va a utilizar ssh para las conexiones al depósito central es importante que el directorio contenga permisos de escritura de grupo de forma que los integrantes de éste puedan acceder al depósito. Para cambiar los permisos escriba una línea como la siguiente (*`deposito`* es el nombre del directorio en el cual se ubica su depósito:

```
[is25201@naoma is25201]$chmod g+wdeposito
```

Es importante anotar que todos los usuarios que desean acceder al depósito deben pertenecer al grupo de la persona dueña del depósito central.

En caso de querer acceder el depósito central usando ssh, estos pasos serán suficientes y podrá continuar a Capítulo 3>. A continuación describimos los pasos siguientes en el caso del protocolo pserver.

2.2. Creación del archivo de usuarios

Se indicará a continuación la forma de crear el archivo `passwd`, cuál permite registrar a los usuarios que tienen acceso al depósito central de datos. Para ello hay que seguir los siguientes pasos:

- Dentro del directorio previamente definido [2], ubíquese dentro del subdirectorio `CVSROOT`. Esto corresponde en la interfaz de comandos, a la siguiente línea.

  ```
  [is25201@naoma is25201]$ cd ~/deposito/CVSROOT
  ```

- Ahora, es necesario definir los usuarios que pueden acceder a dicho depósito central, lo cual lo hacemos en dos pasos:

  1. Creando el primer usuario [3], de la siguiente forma:

     ```
     [is25201@naoma CVSROOT]$htpasswd -c passwd jquiroga
     ```

     Luego digite el *password* correspondiente a este usuario.

     El comando **htpasswd -c** crea un archivo `passwd` en el cual se guarda el *password* del usuario *`jquiroga`*.

2. Para los demás usuarios se omite la opción `-c`, obteniéndose por ejemplo:

   ```
   [is25201@naoma CVSROOT]$  htpasswdpasswd jdavila
   ```

   Y a continuación se escribe el *password* para cada uno de los nuevos miembros.

- Es necesario editar el archivo `passwd` agregando al final de cada línea el nombre de la cuenta.

  En nuestro caso el archivo luce de la siguiente forma:

  ```
  jquiroga:m14lm5:is25201
  jdavila:m14lm5:is25201
  anonymous::is25201
  ```

  Note que hemos creado un usuario `anonymous` sin contraseña pues planeamos que éste sea un usuario de sólo lectura. Del archivo `passwd` y de sus permisos depende la seguridad del sistema entero, la cual reposa en manos del usuario encargado del depósito, cualquier cambio en este archivo puede producir que cualquier persona entre como un usuario que ella cree y tener permisos de *root*. Si su intención es crear un usuario anónimo con permiso de solo lectura, siga al siguiente paso.

- Una vez definidos los usuarios del sistema, es necesario definir cuáles de éstos tendrán permiso de sólo lectura, la forma más fácil de hacer ésto es crear un archivo `readers` en el directorio `CVSROOT` dentro de su directorio de depósito central. Dicho archivo contiene la lista de usuarios con permiso de sólo lectura separados por *Enter*. A continuación presentamos el archivo `readers` de este ejemplo.

  ```
  anonymous
  ```

2.3. Información para administradores

Esta sección sólo es necesaria para el admistrador de la máquina, así que en caso de no tener permisos de usuarios, solicite a su administrador que siga las siguientes instrucciones:

2.3.1. Si dispone de inetd[4] (Redhat 6.2 y anterior)

- Es necesario que en el archivo `/etc/inetd.conf` agregue una linea como la siguiente:

```
cvspserver   stream   tcp   nowait   root   /usr/bin/cvs
cvs -allow-root= /home/is25201/deposito pserver
```

  Note que es necesario cambiar la línea */home/is25201/deposito* por la ruta donde se encuentra el depósito que se acaba de crear. De otro lado *pserver* indica el protocolo mediante el cual se conectan los clientes al depósito central.

- Posteriormente es necesario bajar y subir el servicio **S20inetd** para lo cual son necesarias las siguiente líneas:

```
[root@naoma /]$ cd /etc/rc5.d
[root@naoma rc5.d]$ S20inetd restart
```

2.3.2. Si dispone de xinetd[5] (Redhat 7.0 y posterior)

- Es necesario crear un archivo en`/etc/xinetd.d/cvspserver` con la la siguiente información:

```
# default: on
# description: The cvs allow the creation of remote repositories
service cvspserver
{
disable = no
socket_type = stream
protocol = tcp
wait = no
passenv =
user = root
server = /usr/bin/cvs
server_args     = -allow-root=/home/is25201/deposito -f pserver
```

```
}
```

Note que es necesario cambiar la linea */home/is25201/deposito* por la ruta donde se encuentra el depósito que se acaba de crear. De otro lado *pserver* indica el protocolo mediante el cual se conectan los clientes al depósito central.

- Posteriormente es necesario bajar y subir el servicio **S50xinetd** para lo cual son necesarias las siguiente líneas:

```
[root@naoma /]$ cd /etc/rc5.d
[root@naoma rc5.d]$ S50xinetd restart
```

2.3.3. Información para la distribución Debian

En caso de que no se tenga en su distribucion de GNU/Linux instalado previamente el paquete cvs, tenga en cuenta las siguientes instrucciones para Debian 2.2, es necesario usar el paquete cvs (por ejemplo desde el program dselect). Para configurar cvs(después de instalado) ejecutar dpkg-reconfigure-cvs, la interfaz guiada por menús le solicitará el nombre del directorio donde desea el depósito y si lo desea activará `pserver` (modificando `/etc/inetd.conf`).

Notas

1. En nuestro caso dicho servidor es naoma@uniandes.edu.co
2. En nuestro caso `~/deposito`
3. Dicho nombre no tiene que ver con el *login*, etc. Es simplemente un nombre predeterminado; sin embargo por claridad use los nombres de las personas que tendrán acceso a tal depósito.
4. Demonio de servicios de Internet, inicia los programas asociados a servicios de Internet cuando el sistema inicia.

5. Este es un demonio de servicios de Internet extendido, que inicia los programas asociados a servicios de Internet cuando el sistema inicia.

Capítulo 3. Configuración del cliente en Windows

A continuación describirmemos el proceso de configuracion en el cliente, asi como el conjunto de comandos necesarios para crear un espacio de trabajo y hacer las operaciones usuales, con el depósito central.

3.1. Definición del depósito central

- En este caso usaremos como ssh al que viene con cygwin (http://cygwin.com). Por lo tanto es necesario que para ejecutar wincvs, lo hagamos desde cygwin. Para ello una vez dentro de la línea de comandos cygwin desplacése hasta el directorio donde se encuentra wincvs y ejecútelo. En nuestro caso esto se hace de la siguiente forma

  `Administrador@sistemas141~:` **`cd`**
  `/cygdrive/d/Software/GNU/Wincvs`
  `Administrador@sistemas141~:` **`wincvs &`**

  Desdewincvs realice Admin→Preferences (Alternativamente puede usar **Ctrl-F1**)

- En la pestaña General en la sección Enter the CVSROOT digite una línea como la siguiente:

  `is25201@naoma.uniandes.edu.co:/home/is25201/deposito/`

  Tenga en cuenta que debe modificarse dicha línea de tal forma que se utilice un usuario válido en el servidor. (En nuestro caso es *`is25201@naoma.uniandes.edu.co`*), adicionalmente se debe incluir la ruta completa desde la raíz de la máquina, donde se ubica el depósito central (En nuestro caso es *`/home/is25201/deposito`*)

- En la sección Authentication seleccione la opción correspondiente SSH server, adicionalmente active la opción.

En caso de querer usar la autenticación por medio de pserver, es necesario que la línea de CVSROOT sea como la siguiente **`:pserver:`*`anonymous@naoma.uniandes.edu.co`*`:`** ***`/home/is25201/deposito`***. Adicionalmente en la sección Authentication seleccione la opción correspondiente "passwd" file on the cvs server

Configuración de la variable `CVSROOT` en el cliente

- En la pestaña WinCvs es necesario definir el directorio donde se realizará la extracción. En nuestro caso tendremos una línea como la siguiente **`c:\temp`**

3.2. Ingreso al sistema.

En caso de utilizar pserver, es necesario comenzar una sesión con el depósito central (es decir *login* pra lo cual es necesario que realice los siguientes pasos. En caso de utilizar ssh por favor continue a Sección 3.3>

- Realice Admin→Login y posteriormente coloque su *password* correspondiente (ver Sección 2.2>)
- Verifique que en la parte inferior de la ventana se obtenga el siguiente mensaje

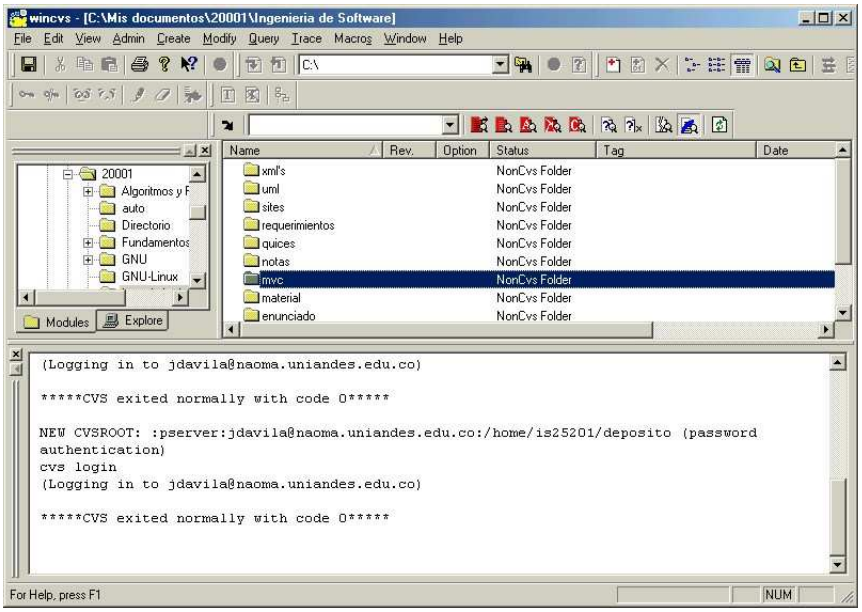
```
*** CVS exited normal with code 0
```

Una vez haya obtenido dicho mensaje podrá realizar las acciones normales sobre cvs.

3.3. Creación de un espacio de trabajo

A continuación crearemos un espacio de trabajo, usando el archivo de ejemplo `FirstMVC`, para ello realice los siguiente

- Baje el archivo (FirstMVC.zip) y descomprímalo en un directorio adecuado.
- Ubíquese dentro del directorio donde descomprimió el archivo usando el explorador de wincvs.



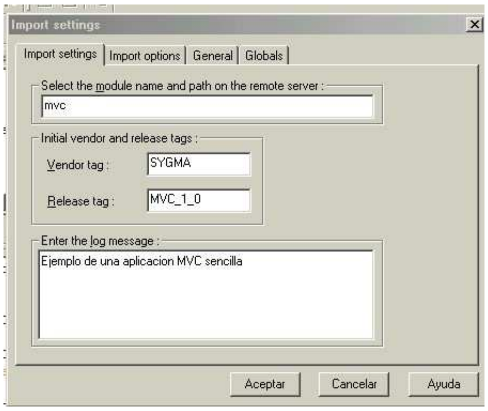
Directorio del archivo descomprimido

- Vaya al menú Create, al submenú Import Module y seleccione el directorio adecuado (es decir aquel donde está descomprimido el archivo), después de lo cual debe obtener una ventana con nombre Import Filter y luego presione continue.

  Resultados de la importación del módulo.

- Se deberá obtener una ventana de Import Settings. En dicha ventana realice lo siguiente:
  - En la sección Select the module name and paht on the remote server, coloque el nombre de como quiere que se llame tal directorio en el depósito central, en nuestro caso lo llamaremos **`mvc`**
  - En la sección Initial vendor and release tags, en el campo Vendor tag coloque el nombre del grupo (en nuestro caso es SIGMA y en el campo Release tag coloque la versión (en nuestro caso es **`MVC_1_0`**).
  - Finalmente, en la sección Enter the log message, coloque una descripción de dicho directorio, en nuestro caso es **`Ejemplo de una aplicación MVC sencilla`**.



Descripción del *checkin*.

- En caso de utilizar ssh, obtendrá una nueva ventana en la cuál le pedirán que acepte la clave del servidor al que se conecte y luego tendrá que digitar su clave. Este tipo de situación se repetirá cada vez que haga una operación con el depósito central, pero puede modificarse usando como se describe en este enlace (http://wincvs.org/ssh), sin embargo este tipo de instrucciones no sirven en el caso de Window 2000™. Deberá obtener el mensaje:

```
No conflict created by this import
```

3.4. Extracción de un módulo

Entenderemos por *checkout* al proceso mediante el cuál extraemos una copia local de un módulo del depósito central, con el fin de posteriormente podamos editarla.

3.4.1. Selección del módulo

Para hacer *checkout* de un módulo realice el siguiente conjunto de instrucciones

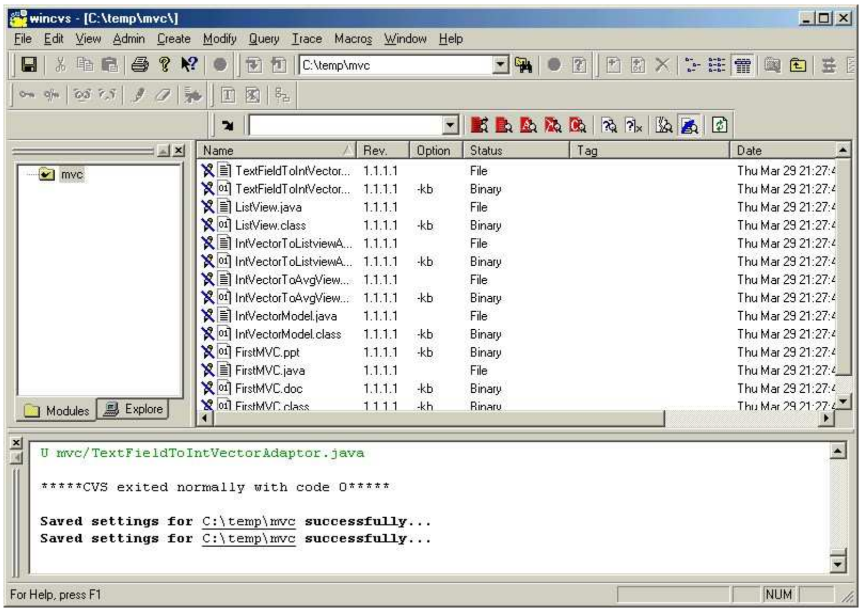
- Seleccione Create→Checkout Module.
- En la pestaña Checkout Settings, en la sección Enter the module name and path on the server coloque el nombre del módulo que desea hacer*checkout*, en nuestro caso **`mvc`**.
- En la sección Local Folder to Checkout, seleccione el directorio donde desea extraer dicho módulo, en nuestro caso, **`c:\temp\mvc`** .
- Deberá obtener el siguiente mensaje

```
***CVS exited with normal code ***
```

3.4.2. Edición de los archivos del módulo.

Es necesario tener la información del módulo que se ha extraído, en el directorio local con el ánimo de poder observar la información de éste, para ello:

- Realice View→Browse Location→Change y posiciónese sobre el directorio sobre el que acabó de hacer *checkout*, obteniendo en una ventana la información de los archivos de cvs.

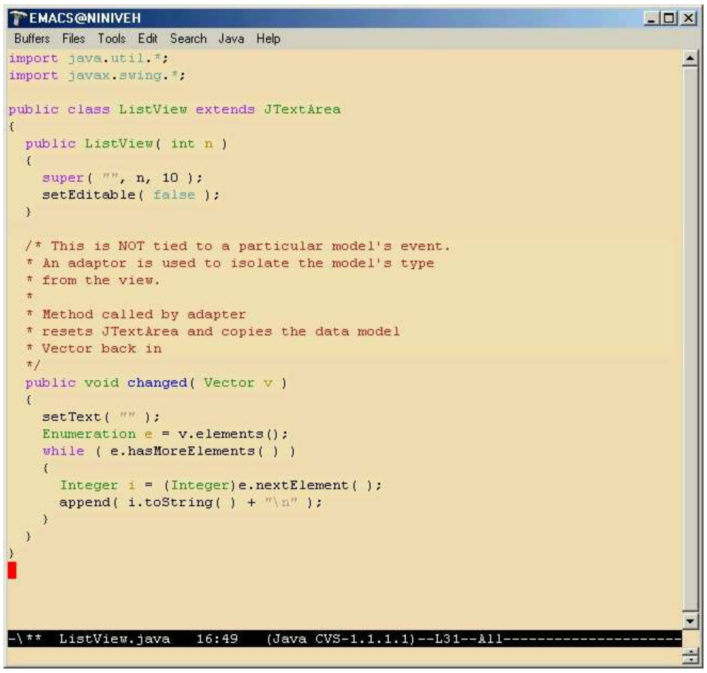


Elección de archivos de *checkout*.

- Para que pueda editar los archivos en su editor de texto favorito[1], seleccione el archivo y hagaTrace→Edit→Select, o haga click sobre el icono del lapiz.

  En nuestro caso, editaremos el archivo `ListView.java`.

- Modifique dicho archivo con su editor de texto, arreglando el estándar de codificación.



Edición del archivo.

- Una vez grabados los cambios sobre tal archivo, usted observará que en wincvs el archivo correspondiente se presenta con color rojo, indicando que tal copia local, no se ha actualizado en el depósito central.

3.5. Actualización del depósito central.

Una vez modificado nuestro archivo deseamos actualizarlo en el depósito central, tal acción recibe el nombre de *commit* o *checkin*.

Sin embargo dicho proceso puede tener variantes de acuerdo a si tenemos la versión más actualizada del archivo que está en el depósito central. A continuación describimos las variantes en cada caso.

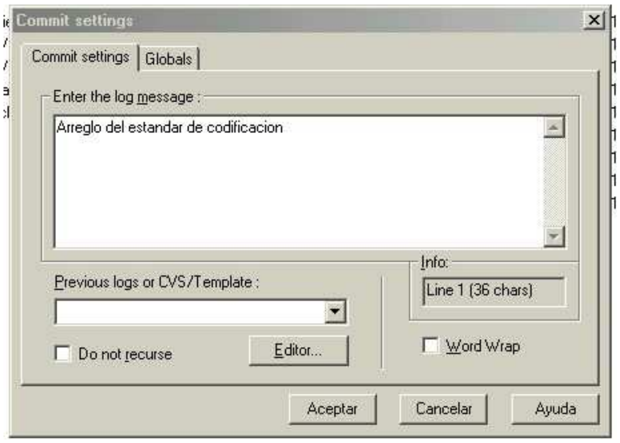
3.5.1. Actualizacion en ausencia de conflictos de versiones

La actualización de un archivo en el depósito central no tendrá conflictos, en caso de que tengamos en nuestro espacio de trabajo la última versión del módulo del depósito central.

Para hacer dicha actualización debemos realizar lo siguiente:

- Seleccionar el o los archivos sobre los que se desear realizar *commit*.
- Para actualizar en el depósito central hay que realizar Modify→Commit Selection (o simplemente **Control**-**M**).

  En la pestaña Commit Settings, en la sección Enter Log Settings, indique la razón del cambio, es decir, **`Arreglo del estándar de codificación`**.

  commit del archivo.

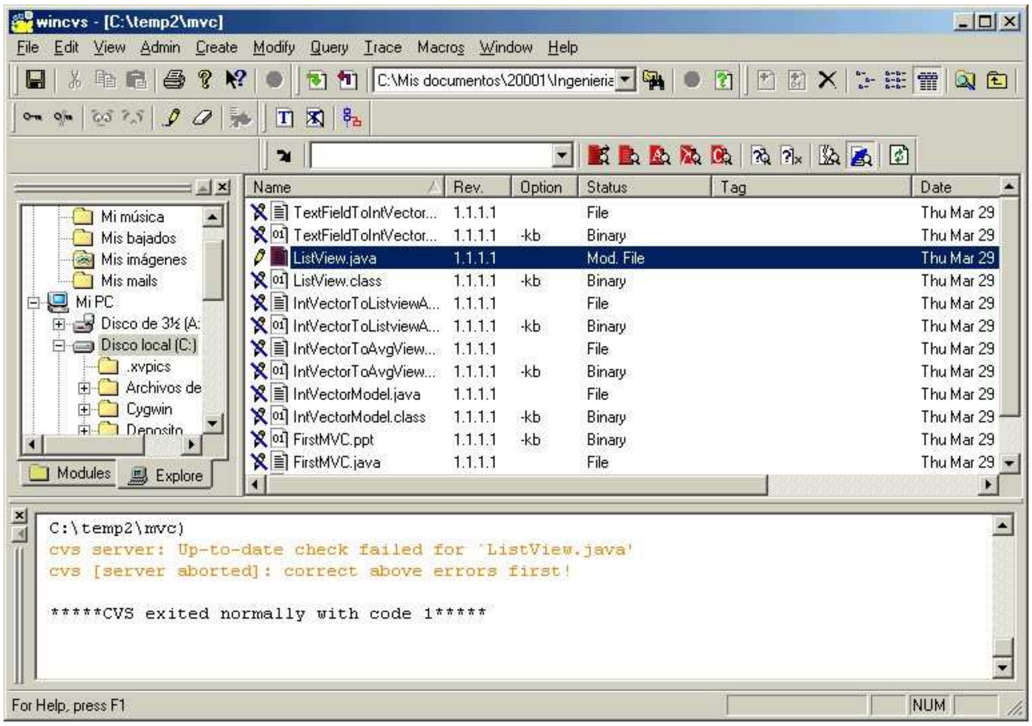
Esto actualiza su depósito y verá que su archivo aparece de nuevo con estatus de poder ser editado, adicionalmente cambia su versión a 1.2.

3.5.2. Actualización en caso de conflictos de versiones

Suponga que se desea hacer *commit* de su archivo, sin embargo otra persona de que su cambio fuera envíado, ha hecho *checkin*, creando una nueva versión y por ende dejando que los cambios que usted ha hecho sobre su archivo desactualizados.

En caso de que intente hacer *commit* de este archivo, obtendrá el siguiente error:

```
Up to date checked failed for ListView.java
```

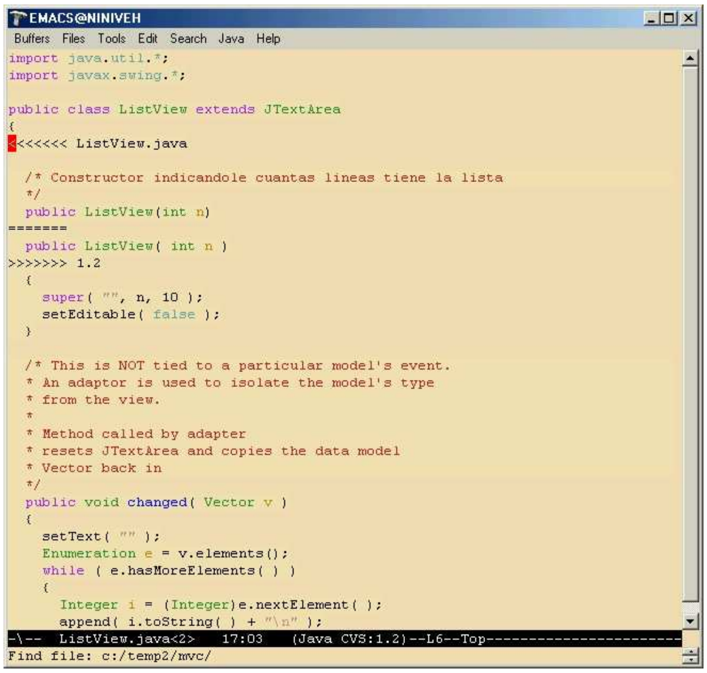


Error al hacer *commit* del archivo.

Por ende tendrá que realizar el siguiente conjunto de pasos:

- Tiene que actualizar el archivo de su espacio de trabajao, para lo cuál realice Modify→Update Selection y luego de Aceptar.
- El archivo se marca con una C, indicando que no hubo problemas en el mezclado de dichos archivos.

- Al editar el archivo se obtiene un archivo con las modificaciones hechas a ambas versiones, por lo que se debe decidir que partes de ambas se desean mantener.
- Grabe las modificaciones que decidió mantener y ahora puede hacer *commit* de tal archivo, creándose una nueva versión.



Edición del archivo con dos versiones.

3.6. Uso de los *tag's*

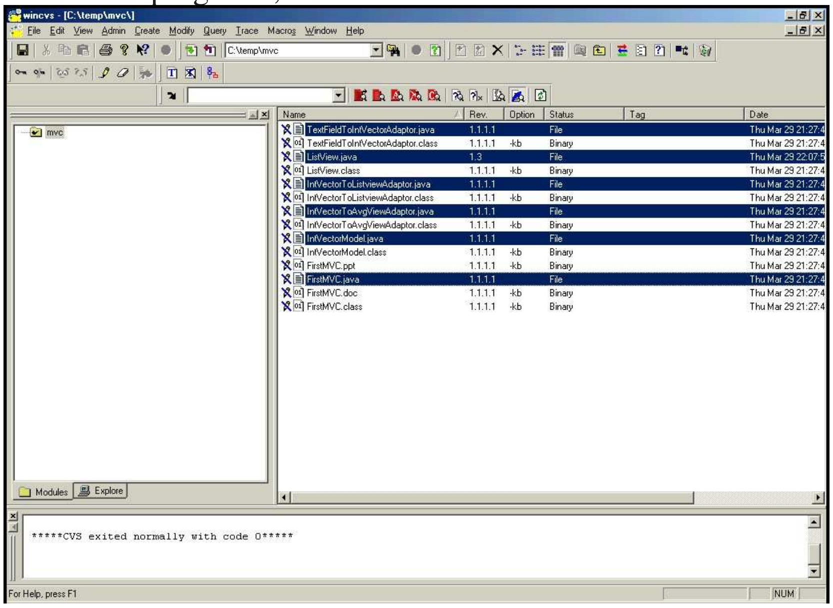
Mediante un *tag* se puede nombrar un conjunto de archivos relacionados a través de un nombre específico, facilitando el uso de éstos. Usando dicho nombre se pueden extraer posteriormente este conjunto de archivos.

A continuación describimos las actividades relacionadas con el uso y creación de dichos *tag's*.

3.6.1. Creacion de los *tag's*

Siga los siguientes pasos:

- Seleccione los archivos y/o directorios relacionados, en nuestro caso los archivos fuentes del programa, es decir todos los archivos con extension `.java`

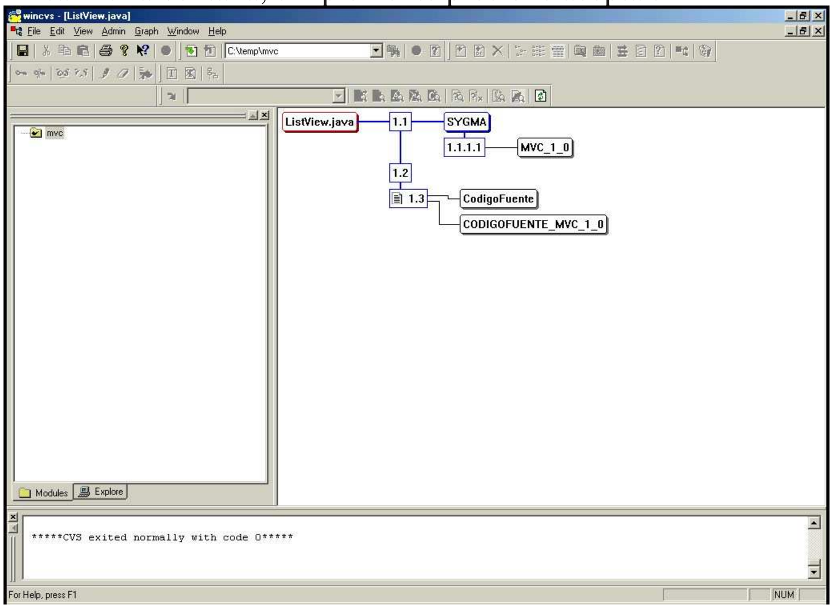
  Escogiendo archivos del *tag*.

- Realice Create→Create a tag by module.
- En la pestaña Create tag settings, en la sección Enter the tag name and which module to tag entre en el campo New Tag Name el nombre del tag deseado (en nuestro caso **`CodigoFuente`**) y en el campo Module to tag el nombre del módulo actual, es decir **`mvc`**.

3.6.2. Observación de los *tag's* disponibles

Para ver los *tag's* que existen sobre un archivo debe hacer lo siguiente

- Seleccione el archivo del cuál desea ver los *tag'* asociados, en nuestro caso escogeremos **`ListView.java`**.
- Realice Query→Graph Selection, o alternativamente **Ctrl-G**
- Al salir una ventana, simplemente presione aceptar.



tags de un archivo.

3.6.3. Extracción de los archivos asociados con un *tag*

Para crear un espacio de trabajo con archivos asociados a un *tag* debe hacer lo siguiente:

- En la pestaña Checkout Settings, en el campo Enter the module name and path on the server, coloque el nombre del módulo (en nuestro caso es **`mvc`**).
- En la pestaña Checkout Settings, en el campo Local folder to checkout to, dé el nombre del directorio donde desea extraer los archivos, por ejemplo `c:\temp\CodigofuenteMVC`.

- En la pestaña Checkout Options, en el campo By Revision/Tag/Branch seleccione o escriba el nombre del tag.
- Por último presione en aceptar.

3.7. Salida de la sesión

Por último, si uso el protocolo pserverno olvide salirse de la sesión de wincvs, para lo cual realice Admin→logout

Notas

1. es decir, emacs

Índice

creación, 23
actualización del depósito central, 19
administración del servidor, 8
Archivo de passwords, 7
autenticación de usuarios, 12
checkin, 19, 20, 20
checkout, 16
commit, 19, 20, 20
conexión al depósito central, 12
conflictos de versiones, 20
copia local, 16
Creación de usuarios, 7
CVSROOT, 6
directorio local, 12
edición, 17
espacio de trabajo, 14
htpasswd, 7
inetd.conf, 8
ingreso, 13
Inicialización del depósito central, 6
login, 13
logout, 25
módulo, 14
passwd, 7
S20inetd, 8
S56xinetd, 8
salida de la sesión, 25
Seguridad, 7
servidor cvs, 6
servidor Linux, 6
Sólo lectura, 7
tag, 22
extración de archivos, 24
tags de un archivo., 23
wincvs, 12
xinetd, 8